

PRIVACY COMPUTING TASK EXECUTION METHOD, STORAGE MEDIUM, AND COMPUTING DEVICE

TECHNICAL FIELD

[0001] One or more embodiments of this specification relate to the field of computer technologies, and in particular, to a kubernetes cluster-based privacy computing task execution method, a privacy task controller in a kubernetes cluster, and a centralized scheduler in a kubernetes cluster.

BACKGROUND

[0002] Kubernetes is briefly referred to as k8s, and is an open-source and industrial container orchestration platform. Currently, emergence of k8s promotes popularization and landing of popular technologies such as a microservice architecture, so that development, operation and maintenance, and delivery become simpler. Although the standard k8s has provided many powerful functions, as application service scenarios become diversified, the standard k8s cannot satisfy a requirement, especially in some special application scenarios. Therefore, secondary development is started based on a k8s base.

[0003] Because a plurality of institutions usually need to jointly participate in a privacy computing task, how to ensure that a task can be efficiently scheduled between different institutions is a relatively popular research field in the industry. Currently, the standard k8s cannot satisfy a specific scenario of scheduling a multi-party privacy computing task. An embodiment of this specification provides a pioneering multi-party privacy computing task scheduling solution, to fill in a gap in this aspect.

SUMMARY

[0004] One or more embodiments of this specification describe a privacy computing task execution method, a storage medium, and a computing device, so that cross-institution pod scheduling can be implemented, scheduling is efficient, and resource utilization is high.

[0005] According to a first aspect, a privacy computing task execution method is provided, related to a privacy task controller and a centralized scheduler in a kubernetes cluster. The method includes: creating, by the privacy task controller, a corresponding task resource group based on description information of a submitted privacy computing task, where information of the task resource group includes information of a plurality of participants of the privacy computing task and a threshold of a quantity of participants that satisfy a scheduling requirement; creating, by the privacy task controller in each of namespaces of the plurality of participants, a task pod used to run the privacy computing task; learning, by the centralized scheduler through watching, that the task pod is created, and obtaining the information of the task resource group based on the task pod; and when it is determined that a quantity of participants that satisfy the scheduling requirement is not less than the threshold of the quantity of participants, correspondingly scheduling, by the centralized scheduler, the task pod to a computing node of the participant that satisfies the scheduling requirement.

[0006] In an embodiment, after the learning, by the centralized scheduler through watching, that the task pod is created, and obtaining the information of the task resource group based on the task pod, the method further includes: obtaining, by the centralized scheduler, node resource information of each of the plurality of participants, where the node resource information is registered in the kubernetes cluster; and determining, based on the node resource information, whether a corresponding participant satisfies the scheduling requirement.

[0007] In an embodiment, the information of the task resource group further includes duration in which a participant reserves a resource; and after the learning, by the centralized scheduler through watching, that the task pod is created, and obtaining the information of the task resource group based on the task pod, the method further includes: when it is determined that a timing reaches the duration and the quantity of participants that satisfy the scheduling requirement is less than the threshold of the quantity of participants, releasing, by the centralized scheduler, a successfully reserved resource of the participant that satisfies the scheduling requirement.

[0008] Further, in a specific embodiment, after the releasing a successfully reserved resource of the participant that satisfies the scheduling requirement, the method further includes: performing, by the centralized scheduler, a next round of scheduling for the privacy computing task.

[0009] In addition, in a specific embodiment, the information of the task resource group further

includes a lifecycle of the task resource group, and the kubernetes cluster further includes a task resource group controller; and the method further includes: updating, by the task resource group controller, a state of the task resource group to "failed" when it is determined that the lifecycle ends and scheduling of the privacy computing task is not completed.

5 **[0010]** In a more specific embodiment, after the updating, by the task resource group controller, a state of the task resource group to "failed", the method further includes: after the privacy task controller learns through watching that the state of the task resource group is "failed", updating, by the privacy task controller, a task state of the privacy computing task to "failed", and/or deleting the task resource group and the task pod.

10 **[0011]** In an embodiment, after the correspondingly scheduling, by the centralized scheduler, the task pod to a computing node of the participant that satisfies the scheduling requirement, the method further includes: updating, by the centralized scheduler, the state of the task resource group to a successful state.

15 **[0012]** In an embodiment, the description information of the privacy computing task includes at least one of the following: information of the plurality of participants, the threshold of the quantity of participants, or a participant scheduling requirement.

20 **[0013]** According to a second aspect, a privacy task controller in a kubernetes cluster is provided, including: a task resource group creation module, configured to create a corresponding task resource group based on description information of a submitted privacy computing task, where information of the task resource group includes information of a plurality of participants of the privacy computing task and a threshold of a quantity of participants that satisfy a scheduling requirement; and a pod creation module, configured to create, in each of namespaces of the plurality of participants, a task pod used to run the privacy computing task, where the kubernetes cluster further includes a centralized scheduler, configured to: learn through watching that the task pod is created, obtain the information of the task resource group based on the task pod, and when
25 it is determined that the quantity of participants that satisfy the scheduling requirement is not less than the threshold of the quantity of participants, correspondingly schedule the task pod to a computing node of the participant that satisfies the scheduling requirement.

30 **[0014]** According to a third aspect, a centralized scheduler in a kubernetes cluster is provided. The kubernetes cluster further includes a privacy task controller; the privacy task controller is configured to: create a corresponding task resource group based on description information of a

submitted privacy computing task, where information of the task resource group includes information of a plurality of participants of the privacy computing task and a threshold of a quantity of participants that satisfy a scheduling requirement; and create, in each of namespaces of the plurality of participants, a task pod used to run the privacy computing task; and the
5 centralized scheduler includes: a resource group information obtaining module, configured to: learn through watching that the task pod is created, and obtain the information of the task resource group based on the task pod; and a pod scheduling module, configured to: when it is determined that a quantity of participants that satisfy the scheduling requirement is not less than the threshold of the quantity of participants, correspondingly schedule the task pod to a computing node of the
10 participant that satisfies the scheduling requirement.

[0015] According to a fourth aspect, a computer-readable storage medium is provided. The computer-readable storage medium stores a computer program, and when the computer program is executed on a computer, the computer is enabled to perform the method in the first aspect.

[0016] According to a fifth aspect, a computing device is provided, and includes a memory
15 and a processor. The memory stores executable code, and when the processor executes the executable code, the method in the first aspect is implemented.

[0017] According to the privacy computing task execution method, the privacy task controller, and the centralized scheduler that are disclosed in the embodiments of this specification, cross-institution pod scheduling can be performed in a multi-party privacy computing task execution
20 scenario, and cross-institution scheduling efficiency and resource utilization are effectively improved.

BRIEF DESCRIPTION OF DRAWINGS

[0018] To describe the technical solutions in the embodiments of the present invention more clearly, the following briefly describes the accompanying drawings required for describing the
25 embodiments. Apparently, the accompanying drawings in the following description show merely some embodiments of the present disclosure, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0019] FIG. 1 is a schematic diagram illustrating a system architecture of scheduling a cross-institution multi-party privacy computing task based on k8s, according to an embodiment;

[0020] FIG. 2 is a schematic flowchart illustrating a privacy computing task execution method, according to an embodiment of this specification;

[0021] FIG. 3 is a schematic diagram illustrating a module structure of a privacy task controller in a k8s cluster, according to an embodiment of this specification; and

5 [0022] FIG. 4 is a schematic diagram illustrating a module structure of a centralized scheduler in a k8s cluster, according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0023] The following describes the solutions provided in this specification with reference to the accompanying drawings.

10 [0024] As mentioned above, a plurality of privacy computing participants perform coordinated computing of a privacy computing task (or referred to as a privacy task in this specification), and a final computing result is obtained. For example, the privacy computing task can be a multi-party joint modeling task, each participant holds some training data, and finally each party obtains a secret sharing segment of a model parameter. Because cross-institution (or referred to as cross-
15 participant) pod scheduling is not considered for a k8s native scheduler, a specific scenario of scheduling a multi-party privacy computing task cannot be satisfied. A pod is a minimum deployable computing unit that can be created and managed in k8s, and a service runs in a pod manner.

[0025] Based on the above-mentioned observation and analysis, embodiments of this
20 specification disclose a k8s-based privacy computing task execution solution, so that cross-institution pod scheduling can be implemented, and efficient scheduling and high resource utilization are implemented.

[0026] FIG. 1 is a schematic diagram illustrating a system architecture of scheduling a cross-institution multi-party privacy computing task based on k8s, according to an embodiment. As
25 shown in FIG. 1, an entire system architecture mainly includes two parts: a k8s management cluster on a central side and a k8s node on an institution side.

[0027] The k8s management cluster includes a task controller (Task Controller), a task resource group controller (Task Resource Group Controller), and a centralized scheduler (centralized scheduler) that are related to this embodiment of this specification. The privacy task

controller is mainly configured to convert a submitted task into a resource supported by k8s. Currently, for each task, the privacy task controller correspondingly creates a resource such as a task resource group (Task Resource Group), a pod, or a service in k8s. The task resource group controller is responsible for a lifecycle of the privacy computing task. It should be understood that the task resource group controller can be disposed independently of the privacy task controller, or can be integrated with the privacy task controller. The centralized scheduler includes an extended scheduling plug-in for the privacy computing task, and implements cross-institution scheduling of the privacy computing task by using the plug-in.

[0028] The k8s node is configured to run a k8s pod service. Each institution can include a plurality of homogeneous or heterogeneous nodes.

[0029] The following describes a procedure of a cross-institution privacy computing task scheduling method in the system architecture shown in FIG. 1. FIG. 2 is a schematic flowchart illustrating a privacy computing task execution method, according to an embodiment of this specification. As shown in FIG. 2, the method includes the following steps.

[0030] Step S210: A privacy task controller obtains a submitted privacy computing task.

[0031] For submission of the privacy computing task, in an embodiment, a user can create a privacy task CR in a k8s cluster by using a k8s kubectl tool. It should be noted that one of extended resource types provided by Kubernetes is a CRD whose full name is a CustomResourceDefinition, that is, a custom resource, and a Kubernetes CR represents a specific instance of the Kubernetes CRD. In another embodiment, an application can create a privacy task CR in the k8s cluster by invoking the k8s API. In this way, the privacy computing task can be submitted.

[0032] Correspondingly, the privacy task controller can obtain description information of the privacy computing task. In an embodiment, task description information includes information of each of a plurality of participants of the privacy computing task, for example, a participant identifier. In another embodiment, task description information includes a threshold of a quantity of participants specified for the privacy computing task, referred to as a minimum quantity of participants. In still another embodiment, task description information includes duration in which a participant can reserve a resource. In yet another embodiment, task description information includes a task life cycle parameter. In still yet another embodiment, task description information includes a creation standard of a task pod in a participant, for example, creating a task pod in a single participant.

[0033] After the submitted privacy computing task is obtained, the privacy task controller can perform step S220, to create a corresponding task resource group based on the description information of the privacy computing task. In view of this, references can be made to a task resource group CR shown in FIG. 1.

5 **[0034]** A specific example of a task resource group CR is shown below, and content included in the task resource group is described with reference to the example.

```

1  apiversion: kuscia.secretflow/v1    //Name kuscia.secretflow and version v1 of a
    custom resource API
2  kind: TaskResourceGroup              //A kind of a custom resource is
10 TaskResourceGroup
3  metadata:
4      name: task-lr-1                  //A name of a custom resource is task-lr-1
5  spec:
6      minReservedMember: 2            //A minimum quantity of participants is 2
15 7      resourceReservedSeconds: 30    //Duration in which a participant can reserve a
    resource
8      lifecycleSeconds: 600           //Lifecycle of a custom resource
9      parties:                         //Participant information
10     - domainID: alice                //A namespace identifier of the 1st participant
20 is alice
11     pods:
12     - name: task-lr-1                //A name of a pod created in a namespace alice
    is task-lr-1
13     - domainID: bob                 //A namespace identifier of the 2nd participant
25 is bob
14     pods:
15     - name: task-lr-1                //A name of a pod created in a namespace bob
    is task-lr-1

```

[0035] With reference to the above-mentioned example, it can be learned that the task resource group can include the following several items of content:

[0036] (1) A name (name) is shown in Row 4 in the above-mentioned example, and indicates the name of the task resource group.

[0037] (2) For a minimum quantity of participants (minReservedMember), references can be

made to Row 6 in the above-mentioned example. The minimum quantity of participants indicates that a minimum quantity of participants that have successfully reserved resources needs to be satisfied during scheduling. Only when the quantity of participants that have successfully reserved resources is greater than or equal to the value, a centralized scheduler schedules a task pod to an institution node.

[0038] (3) For duration in which a participant can reserve a resource (resourceReservedSeconds), references can be made to Row 7 in the above-mentioned example. The duration in which a participant can reserve a resource indicates a time in which a participant can reserve a resource. If a quantity of participants that satisfy the scheduling requirement does not reach the minimum quantity of participants within the time or the duration, the reserved resource is released for use by another subsequent task, thereby improving resource utilization.

[0039] (4) For a lifecycle (lifecycleSeconds) of a task resource group, references can be made to Row 8 in the above-mentioned example. The lifecycle of the task resource group indicates a life cycle of a task resource group CR. If scheduling of the task is not completed within such a time range, a resource state of the task resource group CR is set to "failed".

[0040] (5) For information about a participant (parties), references can be made to Row 9 to Row 15 in the above-mentioned example. The information about the participant is information about the plurality of participants of the privacy computing task.

[0041] It should be noted that specific content in the task resource group can be inherited from the description information of the privacy computing task, or can be filled with a default value if the description information of the privacy computing task is default. In an example, for the minimum quantity of participants in the task resource group, if the task description information includes a corresponding specified value, the set value is used as the minimum quantity of participants; otherwise, a total quantity of participants mentioned in the task description information can be used as the minimum quantity of participants. In another example, for the duration in which a participant can reserve a resource and that is included in the task resource group, if the task description information includes corresponding specified duration (for example, 40s), the specified duration is used as a specific duration value; otherwise, default duration (for example, 30s) is used as a specific duration value.

[0042] In the above-mentioned descriptions, the privacy task controller can create a task resource group CR corresponding to the privacy computing task. Then, the privacy task controller

can further perform step S230, to create, in each of namespaces of the plurality of participants, a task pod used to run the privacy computing task.

[0043] A specific example of a task pod is shown below, and content included in a pod resource is described with reference to the example.

```

5          1  apiversion: v1                      //Version v1 of a resource API
          2  kind: Pod                            //A kind of a resource is pod
          3  metadata:
          4  labels:                              //Label information
of a resource
10          5  task-resource-group.kuscia.secretflow: task-lr-1
          6  name: task-lr-1                      //Name of a resource
          7  namespace: alice                    //Namespace in which a
resource is located
          8  spec:
15          9  containers:
          10 - image: secretflow:v1
          11  command:
          12 - /home/admin/bin/secretflow
          13 - -v
20          14 - 4
          15  name: secretflow
          16  resources:                        //Information about a resource required by a pod
          17  requests:
          18  memory: 512Mi
25          19  cpu: 0.5
          20  limits:
          21  memory: 1Gi
          22  cpu: 1
          23  schedulerName: centralized-scheduler //A name of a scheduler is
30 centralized-scheduler
          24 nodeSelector:
          25  kubernetes.io/namespace: alice    //It is indicated that a node to which a
resource is scheduled needs to have a label "kubernetes.io/namespace: alice"

```

[0044] With reference to the above-mentioned example, it can be learned that the pod resource

can include the following several items of content:

[0045] (1) Label information (labels), for example, shown in Row 4 to Row 5, indicates a task resource group CR bound to the pod.

[0046] (2) A name (name), as shown in Row 6, indicates the name of the pod resource.

5 **[0047]** (3) A namespace (namespace), as shown in Row 7, indicates the namespace in which the pod is located, that is, a namespace of a participant institution.

[0048] (4) A node selector (nodeSelector), as shown in Row 24 and Row 25, indicates that an institution node that needs to be scheduled for the privacy computing task needs to have a label "kubernetes.io/namespace: alice".

10 **[0049]** (5) A scheduler name (schedulerName), as shown in Row 23, indicates the name of the scheduler that needs to be used to schedule the pod. The centralized scheduler (centralized-scheduler) is a scheduler that is extended based on k8s in this embodiment of this specification.

[0050] That the privacy task controller creates a pod resource used to execute the privacy computing task is briefly described above. Based on this, in step S240, the centralized scheduler
15 learns through watching that the task pod is created, and obtains information of the task resource group based on the task pod.

[0051] For example, the centralized scheduler learns through watching that a pod for running a task in a namespace of each participant is created, and then obtains information of a corresponding task resource group CR based on label information (labels) of the pod.

20 **[0052]** Then, in step S250, the centralized scheduler attempts to schedule the pod to a node that satisfies a condition.

[0053] Specifically, the centralized scheduler can obtain node resource information of each of the plurality of participants, and then determine, based on the node resource information, whether a corresponding participant satisfies the scheduling requirement.

25 **[0054]** In an embodiment, the node resource information of each participant is registered in the k8s cluster. Therefore, the centralized scheduler can directly obtain node resource information of the plurality of participants by using the list watch mechanism. In this way, compared with a conventional manner of querying, based on an RPC (Remote Procedure Call) or an http request, whether a computing node of each participant satisfies a scheduling requirement, in this manner,
30 scheduling efficiency can be effectively improved.

[0055] In an embodiment, the scheduling requirement can be obtained by the centralized

scheduler from the information of the task resource group CR. In another embodiment, the scheduling requirement can be set by default in the system.

[0056] In an embodiment, the scheduling requirement includes a resource that needs to be reserved by a node. In this case, the centralized scheduler can determine, based on node resource information of a participant (for example, a remaining resource, an unoccupied resource, and an idle resource), whether the participant can provide a node resource that satisfies the scheduling requirement, and if it is determined that the participant can provide a node resource that satisfies the scheduling requirement, the centralized scheduler locks a resource of a node of the participant, thereby successfully reserving a node resource in the participant.

[0057] Further, when determining that the quantity of participants that satisfy the scheduling requirement is not less than the minimum quantity of participants in the task resource group CR, the centralized scheduler correspondingly schedules, for each participant that satisfies the scheduling requirement, the task pod created in the namespace of the centralized scheduler to a computing node that successfully reserves a resource in the participant. For example, the centralized scheduler can further update the resource state of the task resource group CR to a successful state.

[0058] When determining that the quantity of participants that satisfy the scheduling requirement is less than the minimum quantity of participants in the task resource group CR, the centralized scheduler can stop scheduling. Alternatively, duration in which a participant reserves a resource, and/or a scheduling policy of the lifecycle of the task resource group are introduced.

[0059] Specifically, for a participant that has successfully reserved a resource, resource reservation is continued based on the duration (for example, 30s) in which a participant can reserve a resource. If the centralized scheduler determines that the quantity of participants that satisfy the scheduling requirement reaches the minimum quantity of participants in the duration, the task pod is scheduled. Otherwise, if the centralized scheduler determines that a timing (for example, starting from locking a resource) reaches the duration, but the quantity of participants that satisfy the scheduling requirement does not reach the minimum quantity of participants, the resource that is successfully reserved in the participants that meet the scheduling requirement is released. In this way, a reserved resource can be released in a timely manner to be used for another task, to improve resource utilization. Otherwise, a subsequent task keeps waiting, severely affecting task execution efficiency and resource utilization of a computing node of a participant.

[0060] After the resource is released, in an implementation, the centralized scheduler can stop scheduling the privacy computing task. In another implementation, the centralized scheduler can perform a next round of scheduling on the privacy task. Then, the task pod is successfully scheduled in a round of scheduling in a lifecycle; or when it is determined that a lifecycle (for example, 600s) ends and scheduling of the privacy computing task is not completed, the task resource group controller (TaskResourceGroup Controller shown in FIG. 1) in the k8s cluster updates the state of the task resource group CR "failed".

[0061] For example, the task resource group controller can determine, by determining whether the state of the task resource group CR is updated to "succeed", whether scheduling of the privacy computing task is completed. Specifically, if the state is updated to "succeed", scheduling is completed; or if the state is not updated to "succeed", scheduling is not completed.

[0062] Further, the method can further include: After learning through watching that the state of the task resource group is "failed", the privacy task controller updates the task state corresponding to the privacy computing task to "failed", and deletes the created task resource group CR and the created pod resource; and/or after learning through watching that the state of the task resource group is "failed", the centralized scheduler stops related scheduling.

[0063] In conclusion, according to the privacy computing task execution method disclosed in this embodiment of this specification, cross-mechanism pod scheduling can be implemented in a multi-party privacy computing scenario, and efficiency and resource utilization of cross-mechanism scheduling are effectively improved. Details are as follows: (1) The privacy task controller (Task Controller) is newly added, to convert a multi-party privacy computing task into resources such as a task group resource (TaskResourceGroup CR) and a pod for running a task in k8s, and the module collects a task state. (2) The task group resource controller (TaskResourceGroup Controller) is newly added, to manage the lifecycle of the task group resource CR. (3) A k8s scheduler is extended, to implement coordinated scheduling of a task across a plurality of participants (across Namespace). (4) A task resource group CR is newly added based on a k8s CRD. Some parameters related to task scheduling and lifecycle are recorded in the resource.

[0064] ① The extended k8s scheduler determines, based on the minimum quantity of participants (minReservedMember), whether a condition for scheduling the task Pod to the computing node of the participant is satisfied. Because information about the computing node of

the participant is reported, to register the node resource information in the centralized k8s cluster, the extended k8s scheduler does not need to query, by using a conventional RPC or http request cross a plurality of participants, whether a computing node of each participant satisfies the scheduling requirement.

5 **[0065]** ② The extended k8s scheduler can ensure, based on the duration in which a participant can reserve a resource (resourceReservedSeconds), that when the scheduling requirement is not satisfied, the reserved resource is released in a timely manner to be used for another task, to improve resource utilization. Otherwise, a subsequent task keeps waiting, severely affecting task execution efficiency and resource utilization of a computing node of a participant.

10 **[0066]** ③ The newly added task group resource controller manages the life cycle of the task resource group CR based on the life cycle (lifecycleSeconds) of the task resource group CR and the resource state of the task resource group CR.

[0067] ④ The newly added task controller (Task Controller) watches the state of the task resource group CR and the state of the pod, to manage the lifecycle of the task.

15 **[0068]** Corresponding to the privacy computing task execution method, an embodiment of this specification further discloses a module structure of a privacy task controller and a centralized scheduler that are used in the method.

[0069] FIG. 3 is a schematic diagram illustrating a module structure of a privacy task controller in a k8s cluster, according to an embodiment of this specification. As shown in FIG. 3, a privacy
20 task controller 300 includes the following modules:

 a task resource group creation module 310, configured to create a corresponding task resource group based on description information of a submitted privacy computing task, where information of the task resource group includes information of a plurality of participants of the privacy computing task and a threshold of a quantity of participants that satisfy a scheduling
25 requirement; and a pod creation module 320, configured to create, in each of namespaces of the plurality of participants, a task pod used to run the privacy computing task.

[0070] It should be noted that the k8s cluster further includes a centralized scheduler, configured to: learn through watching that the task pod is created, obtain the information of the task resource group based on the task pod, and when it is determined that the quantity of
30 participants that satisfy the scheduling requirement is not less than the threshold of the quantity of participants, correspondingly schedule the task pod to a computing node of the participant that satisfies the scheduling requirement.

[0071] In an embodiment, the privacy task controller 300 further includes a resource deleting module 330, configured to delete the task resource group and the task pod after it is learned through watching that the state of the task resource group is "failed".

[0072] In an embodiment, the privacy task controller 300 further includes a task state updating module 340, configured to update a task state of the privacy computing task to "failed" after it is learned through watching that the state of the task resource group is "failed".

[0073] FIG. 4 is a schematic diagram illustrating a module structure of a centralized scheduler in a k8s cluster, according to an embodiment of this specification. It should be noted that the k8s cluster further includes a privacy task controller; the privacy task controller is configured to: create a corresponding task resource group based on description information of a submitted privacy computing task, where information of the task resource group includes information of a plurality of participants of the privacy computing task and a threshold of a quantity of participants that satisfy a scheduling requirement; and create, in each of namespaces of the plurality of participants, a task pod used to run the privacy computing task.

[0074] As shown in FIG. 4, a centralized scheduler 400 includes the following modules:
a resource group information obtaining module 410, configured to: learn through watching that the task pod is created, and obtain the information of the task resource group based on the task pod; and a pod scheduling module 420, configured to: when it is determined that a quantity of participants that satisfy the scheduling requirement is not less than the threshold of the quantity of participants, correspondingly schedule the task pod to a computing node of the participant that satisfies the scheduling requirement.

[0075] In an embodiment, the centralized scheduler 400 further includes an evaluation module 430, configured to obtain node resource information of each of the plurality of participants, where the node resource information is registered in the kubernetes cluster; and determine, based on the node resource information, whether a corresponding participant satisfies the scheduling requirement.

[0076] In an embodiment, the information of the task resource group further includes duration in which a participant reserves a resource. The centralized scheduler 400 further includes a resource release module 440, configured to: when it is determined that a timing reaches the duration and the quantity of participants that satisfy the scheduling requirement is less than the threshold of the quantity of participants, release a successfully reserved resource of the participant that satisfies the scheduling requirement.

[0077] An embodiment of another aspect further provides a computer-readable storage medium. The computer-readable storage medium stores a computer program, and when the computer program is executed on a computer, the computer is enabled to perform the method described with reference to FIG. 2.

5 [0078] According to an embodiment in still another aspect, a computing device including a memory and a processor is further provided. The memory stores executable code, and when the processor executes the executable code, the method described with reference to FIG. 2 is implemented.

[0079] A person skilled in the art should be aware that in the one or more examples, functions
10 described in the present invention may be implemented by hardware, software, firmware, or any combination thereof. When the functions are implemented by using software, the functions may be stored in a computer-readable medium or transmitted as one or more instructions or code in the computer-readable medium.

[0080] In the above-mentioned specific implementations, the objectives, technical solutions,
15 and beneficial effects of the present invention are further described in detail. It should be understood that the above-mentioned descriptions are merely specific implementations of the present invention, but are not intended to limit the protection scope of the present invention. Any modification, equivalent replacement, improvement, or the like made based on the technical solutions of the present invention shall fall within the protection scope of the present invention.

20

CLAIMS

What is claimed is:

1. A privacy computing task execution method, comprising:

creating, by a privacy task controller in a kubernetes cluster, a task resource group based on
5 description information of a submitted privacy computing task, wherein information of the task
resource group comprises information of a plurality of participants of the privacy computing task
and a threshold of a quantity of participants that satisfy a scheduling requirement;

creating, by the privacy task controller in each of namespaces of the plurality of participants,
a task pod used to run the privacy computing task;

10 learning, by a centralized scheduler in the kubernetes cluster through watching, that the task
pod is created, and obtaining the information of the task resource group based on the task pod; and

when it is determined that a quantity of participants that satisfy the scheduling requirement is
not less than the threshold of the quantity of participants, scheduling, by the centralized scheduler,
the task pod to a computing node of the participant that satisfies the scheduling requirement.

15 2. The method according to claim 1, wherein after the learning, by a centralized scheduler in
the kubernetes cluster through watching, that the task pod is created, and obtaining the information
of the task resource group based on the task pod, the method further comprises:

obtaining, by the centralized scheduler, node resource information of each of the plurality of
participants, wherein the node resource information is registered in the kubernetes cluster; and

20 determining, based on the node resource information, whether a corresponding participant
satisfies the scheduling requirement.

3. The method according to claim 1, wherein the information of the task resource group
further comprises duration in which a participant reserves a resource; and after the learning, by a
centralized scheduler in the kubernetes cluster through watching, that the task pod is created, and
25 obtaining the information of the task resource group based on the task pod, the method further
comprises:

when it is determined that a timing reaches the duration and the quantity of participants that
satisfy the scheduling requirement is less than the threshold of the quantity of participants,
releasing, by the centralized scheduler, a successfully reserved resource of the participant that
30 satisfies the scheduling requirement.

4. The method according to claim 3, wherein after the releasing a successfully reserved resource of the participant that satisfies the scheduling requirement, the method further comprises:
performing, by the centralized scheduler, a next round of scheduling for the privacy computing task.

5 5. The method according to claim 3 or 4, wherein the information of the task resource group further comprises a lifecycle of the task resource group, and the kubernetes cluster further comprises a task resource group controller; and the method further comprises:

updating, by the task resource group controller, a state of the task resource group to "failed" when it is determined that the lifecycle ends and scheduling of the privacy computing task is not
10 completed.

6. The method according to claim 5, wherein after the updating, by the task resource group controller, a state of the task resource group to "failed", the method further comprises:

after the privacy task controller learns through watching that the state of the task resource group is "failed", updating, by the privacy task controller, a task state of the privacy computing task to "failed", and/or deleting the task resource group and the task pod.
15

7. The method according to claim 1, wherein after the scheduling, by the centralized scheduler, the task pod to a computing node of the participant that satisfies the scheduling requirement, the method further comprises:

updating, by the centralized scheduler, the state of the task resource group to a successful
20 state.

8. The method according to any one of claims 1 to 4, 6, and 7, wherein the description information of the privacy computing task comprises at least one of the following: information of the plurality of participants, the threshold of the quantity of participants, or a participant scheduling requirement.

25 9. A privacy computing task execution method, implemented by a privacy task controller in a kubernetes cluster, the method comprising:

creating a task resource group based on description information of a submitted privacy computing task, wherein information of the task resource group comprises information of a plurality of participants of the privacy computing task and a threshold of a quantity of participants
30 that satisfy a scheduling requirement; and

creating, in each of namespaces of the plurality of participants, a task pod used to run the

privacy computing task, wherein

the kubernetes cluster further comprises a centralized scheduler, configured to: learn through watching that the task pod is created, obtain the information of the task resource group based on the task pod, and when it is determined that the quantity of participants that satisfy the scheduling requirement is not less than the threshold of the quantity of participants, schedule the task pod to a computing node of the participant that satisfies the scheduling requirement.

10. A privacy computing task execution method, implemented by a centralized scheduler in a kubernetes cluster, wherein the kubernetes cluster further comprises a privacy task controller; the privacy task controller is configured to: create a task resource group based on description information of a submitted privacy computing task, wherein information of the task resource group comprises information of a plurality of participants of the privacy computing task and a threshold of a quantity of participants that satisfy a scheduling requirement; and create, in each of namespaces of the plurality of participants, a task pod used to run the privacy computing task; and the method comprises:

15 learning through watching that the task pod is created, and obtaining the information of the task resource group based on the task pod; and

when it is determined that a quantity of participants that satisfy the scheduling requirement is not less than the threshold of the quantity of participants, scheduling the task pod to a computing node of the participant that satisfies the scheduling requirement.

20 11. A computer-readable storage medium, wherein the computer-readable storage medium stores a computer program, and when the computer program is executed on a computer, the computer is enabled to perform the method according to any one of claims 1 to 10.

12. A computing device, comprising a memory and a processor, wherein the memory stores executable code, and when the processor executes the executable code, the method according to 25 any one of claims 1 to 10 is implemented.