

CLAIMS

1. A method for scheduling a task, applied to a scheduler in a cluster management platform which is a platform for managing hardware resources in a cluster, wherein the cluster comprises multiple physical nodes, and wherein the method comprises:

5 acquiring, by the scheduler, a current task to be scheduled from a task scheduling queue, acquiring a task, having a priority greater than or equal to a priority of the current task, on each of the multiple physical nodes, and acquiring a corresponding relationship between the each of the multiple physical nodes and the task, having the priority greater than or equal to the priority of the current task, on the each of the multiple physical nodes to form a node-task mapping table
10 of the multiple physical nodes;

 determining, by the scheduler, a target physical node satisfying a preset screening condition for performing the current task according to the mapping table and the preset screening condition, wherein the preset screening condition comprises a matching condition between resources required by the current task and remaining resources on a physical node and a matching condition
15 between a port required by the current task and a port on the physical node; and

 binding, by the scheduler, the current task with the target physical node, and sending binding information to an application programming interface (API) server in the cluster management platform.

2. The method of claim 1, wherein determining, by the scheduler, the target physical node
20 satisfying the preset screening condition for performing the current task according to the mapping table and the preset screening condition comprises:

 screening out, by the scheduler, physical nodes satisfying a first-stage screening condition from the mapping table to form a node group; and

 scoring, according to the mapping table and a second-stage screening condition, the physical
25 nodes of the node group, and screening out a physical node with a highest score to serve as the target physical node;

 wherein the first-stage screening condition is different from the second-stage screening condition;

 wherein the first-stage screening condition comprises the matching condition between the
30 port required by the current task and the port on the physical node, and whether the physical node has a special label; and

wherein the second-stage screening condition comprises the matching condition between the resources required by the current task and the remaining resources on the physical node.

3. The method of claim 1 or 2, wherein acquiring, by the scheduler, the current task to be scheduled from the task scheduling queue, acquiring the task, having the priority greater than or equal to the priority of the current task, on each of the multiple physical nodes, and acquiring a corresponding relationship between the each of the multiple physical nodes and the task, having the priority greater than or equal to the priority of the current task, on the each of the multiple physical nodes to form the node-task mapping table of the multiple physical nodes comprises:

10 acquiring, by the scheduler, a current pod to be scheduled from a pod scheduling queue, acquiring a pod, having a priority greater than or equal to a priority of the current pod, on the each of the multiple physical nodes, and acquiring a corresponding relationship between the each of the multiple physical nodes and the pod, having the priority greater than or equal to the priority of the current pod, on the each of the multiple physical nodes to form a node-pod mapping table of the multiple physical nodes; and

15 wherein binding, by the scheduler, the current task with the target physical node, and sending the binding information to the API server in the cluster management platform comprises:

binding, by the scheduler, the current pod with the target physical node, and sending the binding information to the API server in the cluster management platform.

20 4. A resource sharing use method based on preemptive scheduling, applied to a cluster management platform which is a platform for managing hardware resources in a cluster, wherein the cluster comprises multiple physical nodes, wherein the cluster management platform comprises a scheduler and an application programming interface (API) server, and wherein the method comprises:

acquiring, by the API server, a creation request of a task;

25 in a case where the API server detects that a quota of a tenant to which the task belongs contains a resource matching with a priority of the task, and the matching resource satisfies a creation condition of the task, creating the task according to the creation request;

acquiring, by the scheduler, the task created by the API server to form a task scheduling queue;

30 acquiring, by the scheduler, a current task to be scheduled from the task scheduling queue, acquiring a task, having a priority greater than or equal to a specified priority of the current task, on each of the multiple physical nodes, and acquiring a corresponding relationship between the

each of the multiple physical nodes and the task, having the priority greater than or equal to the priority of the current task, on the each of the multiple physical nodes to form a node-task mapping table of the multiple physical nodes;

5 determining, by the scheduler, a target physical node satisfying a preset screening condition for performing the current task according to the mapping table and the preset screening condition, wherein the preset screening condition comprises a matching condition between resources required by the current task and remaining resources on a physical node and a matching condition between a port required by the current task and a port on the physical node;

10 binding, by the scheduler, the current task with the target physical node, and sending binding information to the API server;

monitoring, by the target physical node, the binding information between the task and the target physical node in the API server, and acquiring the corresponding task based on the monitored binding information to form a task queue;

15 in a case where the target physical node processes a target task to be run in the task queue, acquiring a task list of tasks running on the target physical node;

detecting, by the target physical node, whether remaining resources obtained after the target physical node executes the tasks in the task list satisfy resources required for running the target task;

20 in a case where the resources required for running the target task are not satisfied, sequentially moving, by the target physical node, at least one task having a priority lower than the target task in the task list into a queue to be removed in a priority order from low to high until the remaining resources of the target physical node satisfy the resources required for running the target task, and adopting the target task to preempt the at least one task in the queue to be removed; and

calling, by the target physical node, an execution environment to run the target task.

25 5. The method of claim 4, wherein acquiring, by the API server, the creation request of the task comprises: acquiring, by the API server, a creation request of a pod; and

wherein in a case where the API server detects that a quota of a tenant to which the task belongs contains a resource matching with a priority of the task, and the matching resource satisfies the creation condition of the task, creating the task according to the creation request comprises:

30 in a case where the API server detects that a quota of a namespace to which the pod belongs

contains a quota value of a resource matching with a priority corresponding to the pod, and the quota value satisfies a creation condition of the pod, creating the pod according to the creation request of the pod.

6. The method of claim 4, further comprising:

5 acquiring, by the target physical node, resource usage information every set time interval;
and

10 in a case where the target physical node determines that the resource usage information reaches a preset limitation condition, sequentially moving at least one task in the task list into the queue to be removed in the priority order from low to high until the resource usage information, determined in a case where the target physical node executes tasks in the task list, does not reach the preset limitation condition, and stopping the at least one task in the queue to be removed.

7. The method of claim 4 or 6, wherein the target task is a target pod, the task list is a pod list, and tasks in the task list are pods running on the target physical node.

15 8. A scheduler, located on a cluster management platform which is a platform for managing hardware resources in a cluster, wherein the cluster comprises multiple physical nodes, and wherein the scheduler comprises:

20 a mapping table forming module, which is configured to: acquire a current task to be scheduled from a task scheduling queue, acquire a task, having a priority greater than or equal to a priority of the current task, on each of the multiple physical nodes, and acquire a corresponding relationship between the each of the multiple physical nodes and the task, having the priority greater than or equal to the priority of the current task, on the each of the multiple physical nodes to form a node-task mapping table of the multiple physical nodes;

25 a screening module, which is configured to determine a target physical node satisfying a preset screening condition for performing the current task according to the mapping table and the preset screening condition, wherein the preset screening condition comprises a matching condition between resources required by the current task and remaining resources on a physical node and a matching condition between a port required by the current task and a port on the physical node; and

30 a binding module, which is configured to: bind the current task with the target physical node, and send binding information to an application programming interface (API) server in the cluster management platform.

9. A resource sharing use system based on preemptive scheduling, comprising an application programming interface (API) server, the scheduler of claim 8, and an apparatus for preempting a task; wherein

the API server comprises:

5 a request acquisition module, which is configured to acquire a creation request of a task; and

a task creation module, which is configured to create the task according to the creation request, in a case of detecting that a quota of a tenant to which the task belongs contains a resource matching with a priority of the task, and the matching resource satisfies a creation condition of the task; and

10 the apparatus for preempting the task comprises:

a task list acquisition module, which is configured to acquire a task list of tasks running on a physical node in a case of processing a target task to be run;

a detection module, which is configured to detect whether remaining resources on the physical node satisfy resources required for running the target task;

15 a preemption module, which is configured to: in a case where the detection module detects that the remaining resources on the physical node do not satisfy the resources required for running the target task, sequentially move at least one task in the task list into a queue to be removed in a priority order from low to high until the remaining resources, obtained in a case where the physical node executes remaining tasks in the task list, satisfy the resources required for running
20 the target task, and a priority of each task in the queue to be removed is less than a priority of the target task, and adopt the target task to preempt the at least one task in the queue to be removed; and

a task execution module, which is configured to call an execution environment to run the target task.

25 10. A device, comprising:

one or more processors; and

a storage apparatus for storing one or more programs;

wherein the one or more programs, when executed by the one or more processors, cause the one or more processors to implement the method for scheduling the task of any one of claims 1-

3.

11. A computer-readable storage medium, storing a computer program, wherein the program, when executed by a processor, implements the method for scheduling the task of any one of claims 1-3.

5