

CLAIMS

1. A task scheduling method, comprising:

determining a dependency relationship among multiple operation tasks according to operands corresponding to the multiple operation tasks in an operation task queue; and

scheduling the multiple operation tasks in the operation task queue based on the dependency relationship among the multiple operation tasks,

wherein determining the dependency relationship among the multiple operation tasks according to the operands corresponding to the multiple operation tasks in the operation task queue comprises:

merging M communication operation tasks included in the multiple operation tasks to obtain at least one merged communication operation task, wherein each of the at least one merged communication operation task comprises at least one of the M communication operation tasks, M being an integer greater than or equal to 1; and

determining the dependency relationship among the multiple operation tasks according to operands corresponding to the at least one merged communication operation task and operands corresponding to at least one non-communication operation task in the multiple operation tasks.

2. The method of claim 1, wherein the operands corresponding to the multiple operation tasks comprise a read operand and/or a write operand.

3. The method of claim 1 or 2, wherein determining the dependency relationship among the multiple operation tasks according to the operands corresponding to the multiple operation tasks in the operation task queue comprises:

if a second operation task contains a read operation performed on a write operand of a first operation task, or the second operation task contains a write operation performed on an operand of the first operation task, determining that the second operation task depends on the first operation task;

wherein the first operation task and the second operation task are different operation tasks in the operation task queue.

4. The method of any one of claims 1 to 3, wherein determining the dependency relationship among the multiple operation tasks according to the operands corresponding to the multiple operation tasks in the operation task queue further comprises:

if a second operation task contains a read operation performed on a read operand of a first operation task, determining that there is no dependency between the first operation task and the second operation task.

5. The method of any one of claims 1 to 4, wherein scheduling the multiple operation tasks in the operation task queue based on the dependency relationship among the multiple operation tasks comprises:

determining a scheduling order of the multiple operation tasks based on the dependency relationship among the multiple operation tasks;

allocating memory for a current operation task in the operation task queue;

scheduling, after allocation of the memory is completed, the current operation task to a context corresponding to the current operation task for execution; and

allocating memory for a next operation task of the current operation task according to the scheduling order.

6. The method of claim 5, wherein determining the scheduling order of the multiple operation tasks based on the dependency relationship among the multiple operation tasks comprises:

if there is no dependency between the first operation task and the second operation task in the multiple operation tasks, determining that the first operation task and the second operation task are scheduled in parallel; and/or,

if the second operation task depends on the first operation task, determining that the second operation task is scheduled after the first operation task.

7. The method of claim 6, wherein the operands corresponding to the at least one merged communication operation task comprises:

a set of read operands corresponding to the at least one communication operation task in the at least one merged communication operation task, and/or a set of write operands corresponding to the at least one communication operation task in the at least one merged communication operation task.

8. The method of any one of claims 1 to 7, wherein the operation task queue comprises a first operation task queue and a second operation task queue,

wherein the first operation task queue contains at least one communication operation task in the multiple operation tasks, and the second operation task queue contains at least one non-communication operation task in the multiple operation tasks;

wherein operation tasks in the first operation task queue and the second operation task queue are arranged in a scheduling order determined based on the dependency relationship among the multiple operation tasks.

9. The method of claim 8, further comprising:

recording dependency information between the first operation task queue and the second operation task queue, wherein,

if an operation task in the first operation task queue depends on at least one operation task in the second operation task queue, or if an operation task in the second operation task queue depends on at least one operation task in the first operation task queue, the dependency information comprises information of a last operation task of the at least one operation task; and

wherein scheduling the multiple operation tasks in the operation task queue

based on the dependency relationship among the multiple operation tasks comprises:

scheduling the operation tasks in the first operation task queue and the second operation task queue based on the dependency information between the first operation task queue and the second operation task queue.

10. The method of claim 8 or 9, further comprising:

setting a priority of a memory reclaim operation task to be the highest, wherein the second operation task queue comprises the at least one non-communication operation task except the memory reclaim operation task in the operation task queue.

11. The method of any one claims 1 to 10, wherein before determining the dependency relationship among the multiple operation tasks, further comprising:

determining a context corresponding to each of the multiple operation tasks, wherein the context corresponding to each operation task comprises an abstract resource and an information stream; and

wherein scheduling the multiple operation tasks in the operation task queue based on the dependency relationship among the multiple operation tasks comprises:

scheduling the multiple operation tasks in the operation task queue based on the context corresponding to each of the multiple operation tasks and the dependency relationship among the multiple operation tasks.

12. The method of claim 11, wherein the information stream comprises a Compute Unified Device Architecture (CUDA) stream and/or a host stream.

13. The method of claim 11 or 12, wherein scheduling the multiple operation tasks in the operation task queue based on the context corresponding to each of the multiple operation tasks and the dependency relationship among the multiple operation tasks comprises:

if there is no dependency between at least two operation tasks in the multiple operation tasks and the at least two operation tasks correspond to different abstract resources, scheduling the at least two operation tasks in parallel.

14. The method of any one of claims 11 to 13, wherein scheduling the multiple operation tasks in the operation task queue based on the context corresponding to each of the multiple operation tasks and the dependency relationship among the multiple operation tasks comprises:

if there is a dependency between a third operation task and a fourth operation task in the multiple operation tasks, and an information stream corresponding to the third operation task and an information stream corresponding to the fourth operation task are both a CUDA stream, synchronizing the third operation task and the fourth operation task by calling a first synchronous interface.

15. The method of any one of claims 11 to 14, wherein scheduling the multiple operation tasks in the operation task queue based on the context corresponding to each of the multiple operation tasks and the dependency relationship among the multiple operation tasks comprises:

if there is a dependency between a third operation task and a fourth operation task in the multiple operation tasks, and an information stream corresponding to at least one of the third operation task and the fourth operation task is a host stream, synchronizing the third operation task and the fourth operation task by calling a second synchronous interface.

16. A task scheduling device, comprising:

a dependency relationship determining module, configured to determine a dependency relationship among multiple operation tasks according to operands corresponding to the multiple operation tasks in an operation task queue; and

a scheduling module, configured to schedule the multiple operation tasks in

the operation task queue based on the dependency relationship among the multiple operation tasks,

wherein the dependency relationship determining module comprises:

a merging submodule, configured to merge M communication operation tasks included in the multiple operation tasks to obtain at least one merged communication operation task, wherein each of the at least one merged communication operation task comprises at least one of the M communication operation tasks, M being an integer greater than or equal to 1; and

a third determining submodule, configured to determine the dependency relationship among the multiple operation tasks according to operands corresponding to at least one merged communication operation task and operands corresponding to at least one non-communication operation task in the multiple operation tasks.

17. The device of claim 16, wherein the dependency relationship determining module further comprises:

a first determining submodule, configured to determine that the second operation task depends on the first operation task if a second operation task contains a read operation performed on a write operand of a first operation task, or the second operation task contains a write operation performed on an operand of the first operation task;

wherein the first operation task and the second operation task are different operation tasks in the operation task queue.

18. The device of claim 16 or 17, wherein the dependency relationship determining module further comprises:

a second determining submodule, configured to determine that there is no dependency between a first operation task and a second operation task if the second operation task contains a read operation performed on a read operand of the first operation task.

19. A non-transitory computer-readable storage medium, having stored therein computer programs, the computer program being configured to execute the task scheduling method of any one of claims 1 to 15.

20. A task scheduling device, comprising:

a processor; and

a memory configured to store instructions executable by the processor;

wherein the processor is configured to call the executable instructions stored in the memory to implement the task scheduling method of any one of claims 1 to 15.