CLAIMS

What is claimed is:

1. A computer-implemented method for graphics processing unit (GPU) task execution, comprising:

in a confidential compute architecture, wherein the confidential compute architecture comprises a secure world, a realm world, a non-secure world, and a root world:

configuring, by GPU software in the non-secure world based on a task code and a cache description of a first GPU task that are provided by a user, a stub data structure of a first stub task in a non-secure world segment of a memory, wherein the stub data structure comprises a plurality of cache areas allocated based on the cache description and metadata indicating each cache area;

wherein the configuring a stub data structure of a first stub task comprises:

allocating a code cache area, and storing the task code in the code cache area;

allocating some stub data cache areas based on the cache description, and storing the cache description in a corresponding stub data cache area; and

generating the metadata, wherein the metadata comprises pointers that point to the code cache area and the some stub data cache areas;

in a segment, in the memory, corresponding to a first realm, configuring, by a root monitor in the root world, a real data structure corresponding to the stub data structure, and storing to-be-processed confidential data in the segment;

updating, by the root monitor, a granule protection table (GPT) so that based on an updated GPT table, a target segment is accessible to a GPU and has realm world permission for all other objects, wherein the target segment stores the metadata and the task code; and

modifying, by the root monitor, a target mapping relationship so that the GPU executes the first GPU task by using the target segment and the real data structure.

2. The computer-implemented method of claim 1, wherein the GPU software comprises a GPU driver and a correlation function library.

3. The computer-implemented method of claim 1, further comprising:

before the configuring a stub data structure of a first stub task:

receiving, by the first realm over a secure channel, the to-be-processed confidential data provided by the user; and

receiving, by a host of the non-secure world, the task code and the cache description.

4. The computer-implemented method of claim 3, further comprising:

negotiating, by the first realm, a key with the user based on a key negotiation protocol; and

constructing the secure channel based on the key.

5. The computer-implemented method of claim 1, wherein configuring a real data structure corresponding to the stub data structure comprises:

allocating some real data cache areas corresponding to the some stub data cache areas; and

storing the to-be-processed confidential data in the some real data cache areas.

6. The computer-implemented method of claim 5, wherein the some real data cache areas comprise an input data cache area and a result data cache area, the input data cache area stores the to-be-processed confidential data, and the result data cache area is configured to store an execution result of the first GPU task.

7. The computer-implemented method of claim 5, wherein:

the configuring a stub data structure of a first stub task comprises:

generating a stub GPU page table based on the some stub data cache areas; and

configuring a real cache segment corresponding to the stub data structure comprises:

generating a real GPU page table based on the stub GPU page table and the some real data cache areas.

8. The computer-implemented method of claim 5, wherein the modifying, by the root monitor, a target mapping relationship comprises:

modifying the pointers in the metadata so that the pointers point to the some real data cache areas and the code cache area.

9. The computer-implemented method of claim 7, wherein the modifying, by the root monitor, a target mapping relationship comprises:

modifying GPU memory mapping so that the GPU memory mapping points to the real GPU page table.

10. The computer-implemented method of claim 1, wherein:

the task code and the cache description of the first GPU task that are provided by the user are accompanied by signature information.

11. The computer-implemented method of claim 10, wherein:

before the modifying a target mapping relationship:

verifying, by the root monitor based on the signature information, the task code stored in the code cache area.

12. The computer-implemented method of claim 1, wherein the updating a granule protection table (GPT) comprises:

updating a first version of GPT table so that the target segment belongs to the realm world in the updated first version of GPT table, wherein the first version of GPT table is used by a CPU and some peripherals.

13. The computer-implemented method of claim 12, further comprising:

updating a second version of GPT table so that in the updated second version of GPT table, a segment of the first realm and the target segment are set to an accessible non-secure world memory and no other segment is accessible, wherein the second version of GPT table is used by the GPU to execute a task of the first realm.

14. The computer-implemented method of claim 13, wherein it is set that a GPU memory mapping segment belongs to the root world in the updated first version of GPT table.

15. The computer-implemented method of claim 13, wherein the first version of GPT table

comprises a first GPT table used for the CPU and some second GPT tables used for the some peripherals, wherein the first GPT table and the some second GPT tables point to a same sub-level table by using table descriptors.

16. The computer-implemented method of claim 15, wherein the same sub-level table is constructed for a preset memory segment, and wherein the target segment belongs to the preset memory segment.

17. The computer-implemented method of claim 1, further comprising:

after the first GPU task is executed, restoring the target segment so that the target segment belongs to the non-secure world.

18. A non-transitory, computer-readable medium storing one or more instructions executable by a computer system to perform one or more operations for graphics processing unit (GPU) task execution, comprising:

in a confidential compute architecture, wherein the confidential compute architecture comprises a secure world, a realm world, a non-secure world, and a root world:

configuring, by GPU software in the non-secure world based on a task code and a cache description of a first GPU task that are provided by a user, a stub data structure of a first stub task in a non-secure world segment of a memory, wherein the stub data structure comprises a plurality of cache areas allocated based on the cache description and metadata indicating each cache area;

wherein the configuring a stub data structure of a first stub task comprises:

allocating a code cache area, and storing the task code in the code cache area;

allocating some stub data cache areas based on the cache description, and storing the cache description in a corresponding stub data cache area; and

generating the metadata, wherein the metadata comprises pointers that point to the code cache area and the some stub data cache areas;

in a segment, in the memory, corresponding to a first realm, configuring, by a root monitor in the root world, a real data structure corresponding to the stub data structure, and storing to-be-processed confidential data in the segment;

updating, by the root monitor, a granule protection table (GPT) so that based on an updated GPT table, a target segment is accessible to a GPU and has realm world permission for all other objects, wherein the target segment stores the metadata and the task code; and

modifying, by the root monitor, a target mapping relationship so that the GPU executes the first GPU task by using the target segment and the real data structure.

19. A computer-implemented system, comprising:

one or more computers; and

one or more computer memory devices interoperably coupled with the one or more computers and having tangible, non-transitory, machine-readable media storing one or more instructions that, when executed by the one or more computers, perform one or more operations for graphics processing unit (GPU) task execution, comprising:

in a confidential compute architecture, wherein the confidential compute architecture comprises a secure world, a realm world, a non-secure world, and a root world:

configuring, by GPU software in the non-secure world based on a task code and a cache description of a first GPU task that are provided by a user, a stub data structure of a first stub task in a non-secure world segment of a memory, wherein the stub data structure comprises a plurality of cache areas allocated based on the cache description and metadata indicating each cache area;

wherein the configuring a stub data structure of a first stub task comprises:

allocating a code cache area, and storing the task code in the code cache area;

allocating some stub data cache areas based on the cache description, and storing the cache description in a corresponding stub data cache area; and

generating the metadata, wherein the metadata comprises pointers that point to the code cache area and the some stub data cache areas;

in a segment, in the memory, corresponding to a first realm, configuring, by a root monitor in the root world, a real data structure corresponding to the stub data structure, and storing to-be-processed confidential data in the segment;

updating, by the root monitor, a granule protection table (GPT) so that based on an updated GPT table, a target segment is accessible to a GPU and has realm world permission

for all other objects, wherein the target segment stores the metadata and the task code; and

modifying, by the root monitor, a target mapping relationship so that the GPU executes the first GPU task by using the target segment and the real data structure.