dependency relationship. Based on this, a computing platform allocates the plurality of small tasks to machines of a plurality of institutions by using the task scheduling method provided in this specification, to fully use computational resources of the institutions.

[0033] It should be noted that the above-mentioned application scenario of the privacy-preserving computation platform is only one of a plurality of application scenarios provided in this specification. A person skilled in the art should understand that the task scheduling method and system described in this specification applying to other application scenarios also fall within the protection scope of this specification.

[0034] FIG. 1 is a schematic diagram of an application scenario of a task scheduling system 003 according to some embodiments of this specification. For ease of understanding, a privacy-preserving computation scenario is used as an example for description in the following description. As shown in FIG. 1, the application scenario can include a computing platform 001, a computing node 002, and a task scheduling system 003. The task scheduling system 003 can be configured to schedule a plurality of tasks in the computing platform 001 for computation, to schedule the plurality of tasks to a plurality of machines in the computing node 002.

[0035] The computing node 002 can be computing resources and equipped facilities deployed in an institution. Different computing nodes 002 can be deployed in different institutions. In the schematic diagram shown in FIG. 1, one computer node 002 can correspond to one institution. Privacy-preserving computation is used as an example. Different types of institutions may exist in different application scenarios. For example, in training of an information recommendation model, the institutions can include a merchant with an information recommendation requirement, a media platform, and a third-party data management platform. The media platform can provide rich user portrait data. A plurality of institutions can select, through user data matching, data of a common user as training data for use in subsequent model training. For another example, in training of a credit risk assessment model of a financial institution, the institutions can include a financial institution and a third-party data source institution. There may be one or more third-party data source institutions, which are used to provide high-quality compliant user data. More comprehensive training data can be provided for the risk assessment model by combining the data of the plurality of data source institutions.

[0036] The institution can connect the computing node 002 to the computing platform 001 in a plurality of manners, such as privatized deployment, clustered deployment, or software as a service (SaaS) deployment-free. The computing platform 001 can schedule a computing device in the computing node 002. For ease of description, a quantity of a plurality of computing nodes 002 is defined as M. M is an integer greater than 1. Each computing node 002 can include at least one computing device. Each computing device can be a server or a client. A plurality of computing

(obtaining an execution result of the upstream task F) plus expected execution duration 2, that is, 5. An expected completion moment at which the task C is executed on the machine 1 is the expected start moment 11 plus the expected duration 5, that is, 16. Similarly, an expected start moment of the task C on the machine 2 is 7, expected duration is 6, and an expected completion moment is 13. Therefore, the machine 2 is selected as the target machine. In the fourth round of allocation, the task C is allocated to the machine 2.

[0102] In a fifth round of allocation, the task D and a task E are candidate tasks in the current round of allocation. The earliest idle moment of the machine 1 and the machine 2 is 11. Executable moments of the task D and the task E are respectively 11 and 13. Therefore, the task D is selected as the target task. An expected start moment of the task D on the machine 1 is 11. Expected duration required when the task D is executed on the machine 1 is expected communication duration 0 plus expected execution duration 1, that is, 1. An expected completion moment at which the task D is executed on the machine 1 is the expected start moment 11 plus the expected duration 1, that is, 12. Similarly, an expected start moment of the task D on the machine 2 is 13, expected duration is 3, and an expected completion moment is 16. Therefore, the machine 1 is selected as the target machine. In the fifth round of allocation, the task D is allocated to the machine 1.

[0103] In a sixth round of allocation, the task E is the only candidate task in the current round of allocation, and is also the target task. An expected start moment of the task E on the machine 1 is 13 (the moment 13 at which all upstream tasks are completed is later than an expected idle moment 12 of the machine 1). Expected duration required when the task E is executed on the machine 1 is expected communication duration 3 plus expected execution duration 2, that is, 5. An expected completion moment at which the task E is executed on the machine 1 is the expected start moment 13 plus the expected duration 5, that is, 18. Similarly, an expected start moment of the task E on the machine 2 is 13, expected duration is 4, and an expected completion moment is 17. Therefore, the machine 2 is selected as the target machine. In the sixth round of allocation, the task E is allocated to the machine 2.

[0104] In conclusion, after six rounds of task allocation are performed, a task scheduling table is obtained: {machine 1: task A, task B, and task D; machine 2: task F, task C, and task E}. It can be learned that, after allocation is performed by using the task scheduling method provided in this specification, tasks in an upstream-downstream dependency relationship are allocated to a same machine as far as possible, thereby shortening consumed communication time and improving overall task execution efficiency.

[0105] Another aspect of this specification provides a computer-readable non-temporary storage medium. The computer-readable non-temporary storage medium stores at least one group of executable instructions use to perform task scheduling. When the executable instructions are