

SCHEDULING METHODS, APPARATUSES, ELECTRONIC DEVICES AND STORAGE MEDIA

TECHNICAL FIELD

[0001] The present disclosure relates to the field of computer technology, and in particular, to scheduling methods, apparatuses, electronic devices and storage media.

BACKGROUND

[0002] With the rapid development of artificial intelligence technology, a large number of artificial intelligence companies have been established. While a neural network used in artificial intelligence relies on computing power of a graphics processing unit (GPU), and computing tasks of the GPU are gradually developing from small-scale to large-scale and high-speed network interconnection. In order to meet different requirements of different tasks on the GPU, the academia and the industry have begun to explore GPU virtualization technology.

SUMMARY

[0003] A scheduling scheme is provided in the present disclosure.

[0004] According to an aspect of the present disclosure, a scheduling method is provided, including: obtaining a task processing time of a first task in a current scheduling period; determining whether the first task is overtime based on the task processing time and a first time slice allocated to the first task in the current scheduling period; and adjusting one or more time slices allocated to one or more second tasks in a target scheduling period according to a result of the determination, where the target scheduling period includes the current scheduling period or a next scheduling period of the current scheduling period.

[0005] In one or more alternative embodiments, determining whether the first task is overtime based on the task processing time and the first time slice allocated to the first task in the current scheduling period, includes: determining a first time difference between a deadline of the task processing time and a deadline of the first time slice; and determining, in a case that the first time difference is greater than a first time threshold, that the first task is overtime.

[0006] In one or more alternative embodiments, the first time threshold is determined based on a length of the first time slice.

[0007] In one or more alternative embodiments, adjusting the one or more time slices allocated to the one or more second tasks in the target scheduling period according to a result of the determination, includes: adjusting, in a case of determining that the first task is overtime, a length of the current scheduling period; and adjusting the one or more time slices allocated to the one or more second tasks in the current scheduling period based on the adjusted length of the current scheduling period.

[0008] In one or more alternative embodiments, adjusting the one or more time slices allocated to the one or more second tasks in the current scheduling period based on the adjusted length of the current scheduling period, includes: for each of the one or more second tasks, adjusting a length of a time slice allocated to the second task in the current scheduling period based on the adjusted length of the current scheduling period and a Quality Of Service QOS parameter of the second task to achieve at least one of: maintaining the QOS parameter of the second task at a corresponding preset value, and maintaining a proportion of the time slice of the second task with respect to the current scheduling period at a corresponding preset value.

[0009] In one or more alternative embodiments, adjusting, in a case of determining that the first task is overtime, the length of the current scheduling period, includes: obtaining a

number of overtime third tasks in the current scheduling period, where the third tasks indicate one or more tasks that have been executed and completed in the current scheduling period; determining an extension coefficient for extending the length of the current scheduling period according to the number of overtime third tasks, where the extension coefficient is positively correlated with the number of overtime third tasks; and acquiring the adjusted length of the current scheduling period by extending the length of the current scheduling period based on the extension coefficient.

[0010] In one or more alternative embodiments, the extension coefficient is determined based on an exponential function with the number of overtime third tasks as a variable.

[0011] In one or more alternative embodiments, the method further includes: obtaining a task processing time of a third task in the current scheduling period, where the one or more second tasks include the third task; and performing, in response to that the task processing time of the third task is less than an adjusted length of a time slice of the third task, time compensation on the third task in the current scheduling period.

[0012] In one or more alternative embodiments, performing, in response to that the task processing time of the third task is less than the adjusted length of the time slice of the third task, time compensation on the third task in the current scheduling period, includes: determining a compensation time for the third task based on the task processing time of the third task and the adjusted length of the time slice of the third task; and adding, based on the compensation time, a compensation task for the third task to a scheduling queue of the current scheduling period.

[0013] In one or more alternative embodiments, adding, based on the compensation time, a compensation task for the third task to the scheduling queue of the current scheduling period, includes: adding, based on the compensation time, the compensation task for the third task to an end of the scheduling queue of the current scheduling period.

[0014] In one or more alternative embodiments, adjusting the one or more time slices allocated to the one or more second tasks in the target scheduling period according to a result of the determination, includes: adjusting, in a case of determining that each task in the current scheduling period is not overtime, a length of the next scheduling period of the current scheduling period; and determining the one or more time slices allocated to the one or more second tasks in the next scheduling period based on the adjusted length of the next scheduling period.

[0015] In one or more alternative embodiments, adjusting the length of the next scheduling period of the current scheduling period, includes: reducing the length of the next scheduling period based on a preset reduction coefficient.

[0016] In one or more alternative embodiments, determining the one or more time slices allocated to the one or more second tasks in the next scheduling period based on the adjusted length of the next scheduling period, includes: for each of the one or more second tasks, determining a length of a time slice allocated to the second task in the next scheduling period based on the adjusted length of the next scheduling period and a QOS parameter of the second task to achieve at least one of: maintaining the QOS parameter of the second task at a corresponding preset value, and maintaining a proportion of the time slice of the second task with respect to the next scheduling period at a corresponding preset value.

[0017] In one or more alternative embodiments, the method further includes: obtaining respective historical task processing time of each of historical tasks executed and completed in a preset time period; and determining an initial length of the current scheduling period according to statistical information of the respective historical task processing time.

[0018] According to an aspect of the present disclosure, an apparatus for scheduling is provided, including: an obtaining unit, configured to obtain a task processing time of a first task in a current scheduling period; a determining unit, configured to determine whether the

first task is overtime based on the task processing time and a first time slice allocated to the first task in the current scheduling period; and an adjusting unit, configured to adjust one or more time slices allocated to one or more second tasks in a target scheduling period according to a result of the determination, where the target scheduling period includes the current scheduling period or a next scheduling period of the current scheduling period.

[0019] According to another aspect of the present disclosure, an electronic device is provided, including: a processor; a memory for storing computer readable instructions; wherein the processor is used to call the computer readable instructions stored in the memory to execute the above scheduling method.

[0020] According to another aspect of the present disclosure, a computer readable storage medium with computer program instructions stored thereon is provided, wherein when the computer program instructions are executed by a computer, the above scheduling method is executed.

[0021] According to another aspect of the present disclosure, a computer program including computer-readable instructions is provided, when the computer-readable instructions are executed by a computer, the above scheduling method is executed.

[0022] In the embodiments of the present disclosure, a task processing time of a first task in a current scheduling period can be obtained, then whether the first task is overtime can be determined based on the task processing time and a first time slice allocated to the first task in the current scheduling period, and then one or more time slices allocated to one or more second tasks in a target scheduling period can be adjusted according to a result of the determination. The target scheduling period includes the current scheduling period or a next scheduling period of the current scheduling period. In this way, the time slices allocated to the second tasks in the target scheduling period can be automatically adjusted, so that the

time slice allocated to each second task in the target scheduling period could meet the requirements of quality of service, such that user experience can be improved.

[0023] It should be understood that the above general description and the following detailed description are only exemplary and explanatory, rather than limiting the present disclosure.

[0024] According to the following detailed description of exemplary embodiments with reference to the accompanying drawings, other features and aspects of the present disclosure will become clear.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate examples consistent with the present disclosure, and together with the description, serve to explain the solutions of the disclosure.

[0026] FIG. 1 is a flowchart illustrating a scheduling method according to embodiments of the present disclosure.

[0027] FIG. 2 is a flowchart illustrating an example of extending a time slice allocated to a task according to embodiments of the present disclosure.

[0028] FIG. 3 is a flowchart illustrating an example of reducing a time slice allocated to a task according to embodiments of the present disclosure.

[0029] FIG. 4 illustrates a curve showing the relationship between historical task processing time and cumulative frequency according to embodiments of the present disclosure.

[0030] FIG. 5 is a block diagram illustrating an apparatus for scheduling according to embodiments of the present disclosure.

[0031] FIG. 6 is an exemplary block diagram illustrating an electronic device according to embodiments of the present disclosure.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0032] Various exemplary embodiments, features and aspects of the present disclosure will be described in detail below with reference to the accompanying drawings. The same reference signs in the drawings indicate elements with the same or similar functions. Although various aspects of the embodiments are shown in the drawings, unless otherwise noted, the drawings are not necessarily drawn to scale.

[0033] The dedicated word "exemplary" here indicates "used as an example, an embodiment or an illustration". Herein, any embodiment described as "exemplary" is not necessarily construed as superior to or better than other embodiments.

[0034] The term "and/or" in the present disclosure is merely an association relationship for describing associated objects, and indicates that there may be three association relationships, for example, A and/or B may indicate that there are three cases: A alone, both A and B, and B alone. In addition, the term "one or more" herein indicates any one of a plurality or any combination of at least two of the plurality. For example, including one or more of A, B, and C, can indicate that including any one or more elements selected from a set of A, B, and C.

[0035] In addition, numerous specific details are given in the following specific embodiments to better illustrate the present disclosure. Those skilled in the art should understand that the present disclosure can also be implemented without certain specific details. In some examples, the methods, means, elements, and circuits that are well known to those skilled in the art are not described in detail in order to highlight the gist of the present disclosure.

[0036] In the scheduling scheme provided by the embodiments of the present disclosure, a task processing time of each task on the vGPU (virtual Graphics Processing Unit) in a current scheduling period can be obtained, and then whether a first task is timed out can be determined based on the task processing time of the first task and a first time slice allocated to the first task in the current scheduling period, such that one or more time slices allocated to one or more second tasks in a target scheduling period can be adjusted according to the determined result. In this way, when the first time slice allocated to the first task is too long or too short, the time slice allocated to the second task can be automatically adjusted, such that the time slice of the second task in the target scheduling period can be reasonably configured, and the GPU can flexibly set the allocated time slices while having higher performance, and the resources of the GPU can be effectively used, and the user experience can be improved.

[0037] The GPU, that is, a physical GPU, can be divided into smaller granular units, and each unit can be called a virtual image processing unit. The virtual image processing unit can process tasks by time-division multiplexing the GPU, and each task can include a plurality of task units. When the virtual image processing unit is executing a task, a task unit being executed cannot be suspended. Therefore, when processing time of the virtual image processing unit to execute a task unit is relatively long, the task processing time of the task will exceed the time slice allocated to the task, thereby reducing quality of service (QoS) of other tasks. The quality of service can be an index for evaluating the ability to provide services. In the scheduling scheme provided by the embodiments of the present disclosure, the task processing time of a task can be matched to the time slice allocated to the task, so that the quality of service provided by the scheduling system is guaranteed. The task unit here can be understood as a basic unit of task processing, usually cannot be divided into smaller units, and cannot be suspended during execution. The virtual image processing unit

can be used by the plurality of task units in a time-division multiplexing mode. For example, a task unit can correspond to a kernel function. A kernel function can include a set of instructions, and a length of each kernel function can be the same or different. For the execution of a kernel function, it will not end until all the instructions of the kernel function have been executed.

[0038] A device may include one or more GPUs, and a GPU may include one or more virtual image processing units as well. The scheduling method provided in the present disclosure can be applied to a single virtual image processing unit, and can also be applied to a plurality of virtual image processing units. The scheduling method of each of the plurality of virtual image processing units is similar to the scheduling method of a single virtual image processing unit. For the sake of simplicity, a single virtual image processing unit will be used as an example for description in the following. Those skilled in the art should understand that the present disclosure does not limit the number of virtual image processing units. Similarly, those skilled in the art should understand that the present disclosure does not limit the number of GPUs.

[0039] The technical solutions provided by the embodiments of the present disclosure can be applied to GPU task processing, virtual image processing unit scheduling, time slice adaptive expansion, etc. which is not limited in the embodiments of the present disclosure.

[0040] FIG. 1 is a flowchart illustrating a scheduling method according to embodiments of the present disclosure. The scheduling method may be executed by a terminal device, a server or other types of electronic devices, where the terminal device may be a UE (User Equipment), a mobile device, a user terminal, a terminal, a cellular phone, a cordless phone, a PDA (Personal Digital Assistant), a handheld device, a computing device, an in-vehicle device, a wearable device, etc. In some possible implementations, the scheduling method may be implemented with a processor by invoking computer-readable instructions stored in

a memory. The scheduling method can be applied to schedule a plurality of tasks on the same virtual image processing unit. For each of the plurality of tasks, one or more time slices can be allocated, and the execution of the task can be implemented by executing at least one task unit of the task in the time slices. In the following, description of the scheduling method in the present disclosure will be given with reference to an example in which an execution entity is an electronic device.

[0041] At step 11, obtaining a task processing time of a first task in a current scheduling period.

[0042] In the embodiments of the present disclosure, the electronic device can detect the task processing time of each of the plurality of tasks in the current scheduling period, and the first task can indicate the task currently being processed. The current scheduling period can indicate a scheduling period in which the task currently being processed. In the virtual image processing unit, a task queue includes a plurality of scheduling periods.

[0043] In some examples, in the virtual image processing unit, a scheduling period is relatively short, for example, it can be a few seconds or hundreds of milliseconds. While a period of a task queue is relatively long, for example, it can be several hours or even several days. Since a task in the task queue may need to be executed in a plurality of scheduling periods to be completed, correspondingly, a certain task in a scheduling period described in the present disclosure refers to a part of the task allocated to be executed in the scheduling period, which can be all task units of the task, or part of the task units of the task. A plurality of tasks processed by time division in two task scheduling periods may be the same or different.

[0044] The virtual image processing unit can be used by each of the plurality of tasks in a time-division multiplexing mode. Correspondingly, the plurality of tasks can form a scheduling queue. A task can be an image processing task or a neural network training task.

The present disclosure does not limit the tasks specifically processed. The task processing time of a task can be the total time spent to execute the task in a scheduling period, where the execution of the task can include executing one or more task units of the task.

[0045] At step 12, determining whether the first task is overtime based on the task processing time and a first time slice allocated to the first task in the current scheduling period.

[0046] In the embodiments of the present disclosure, a comparison result can be obtained by comparing the task processing time of the first task with the first time slice allocated to the first task, and whether the first task is overtime can be determined according to the comparison result. The first time slice can be the time allocated to the first task for occupying the virtual image processing unit. A length of a time slice allocated to each task can be the same or different.

[0047] For example, a first task includes a plurality of task units, and a first time slice is allocated to the first task in the current scheduling period for executing at least two task units of the first task. In a process of executing the at least two task units of the first task based on the first time slice in the current scheduling period, it is assumed that the execution starts from the n-th task unit, and when the m-th task unit is executing, the first time slice is expired, but because a task unit cannot be suspended, a next task will not be started and executed until the completion of the m-th task unit. In this case, the task processing time of the first task, that is, the time taken from the start of executing the n-th task unit to the completion of executing the m-th task unit, exceeds the length of the first time slice allocated to the first task, that is, the first task is overtime. m and n are both positive integers, $m \geq n$.

[0048] For example, the length of the task processing time of the first task can be compared with the length of the first time slice to determine whether the length of the task

processing time of the first task is greater than the length of the first time slice. If the length of the task processing time of the first task is greater than the length of the first time slice, it can be determined that the first task is overtime. If the length of the task processing time of the first task is less than or equal to the length of the first time slice, it can be determined that the first task is not overtime.

[0049] For another example, a deadline of the task processing time of the first task can be compared with a deadline of the first time slice, in a case that the deadline of the task processing time of the first task is later than the deadline of the first time slice, it can be determined that the first task is overtime. In a case that the deadline of the task processing time of the first task is earlier than or equal to the deadline of the first time slice, it can be determined that the first task is not overtime.

[0050] In a possible implementation, a first time difference between the deadline of the task processing time of the first task and the deadline of the first time slice can be determined, and then in a case that the first time difference is greater than a first time threshold, the first task can be determined as overtime.

[0051] In this implementation, the first time difference between the deadline of the task processing time of the first task and the deadline of the first time slice can be calculated first, and then the calculated first time difference can be compared with the preset first time threshold. In a case that the first time difference is greater than the first time threshold, it can be determined that the first task is overtime. In a case that the first time difference is less than or equal to the first time threshold, it can be determined that the first task is not overtime.

[0052] Still taking the above example for description, the time slice is expired when executing the m-th task unit, but because a task unit cannot be suspended, the next task will not be started and executed until the completion of the m-th task unit. In this case, the first

task is not directly determined as overtime, but it can be determined that a first time difference between the deadline of the task processing time of the first task, that is, a deadline of the m-th task unit, and the deadline of the first time slice. In a case that the first time difference is greater than the first time threshold, then the first task is determined as overtime.

[0053] In some examples, the first time threshold can be a preset fixed value, for example, the first time threshold may be set to 0.2s, 0.5s, etc.

[0054] In some examples, the first time threshold can be set according to a preset ratio value. For example, the above-mentioned first time threshold can be determined based on the length of the first time slice. For example, the first time threshold can be set to 1% of the length of the first time slice, in this way the first time threshold can be decided by the length of the first time slice, which makes setting the first time threshold more flexible. When the first time slice is relatively short, the first time threshold is relatively small, that is, a condition for overtime determination is relatively harsh; while when the first time slice is relatively long, the first time threshold is relatively large, that is, the condition for overtime determination is relatively loose. For another example, the first time threshold can also be determined based on the length of the current scheduling period, for example, the first time threshold is set to 1% of the length of the current scheduling period.

[0055] Through the first time threshold, the first time difference between the task processing time of the first task and the first time slice can be allowed to be within a reasonable range, so that a process of adjusting a time slice allocated to a second task can converge, rather than adjusting the time slice allocated to the second task too frequently. For example, when the first time threshold is adjusted proportionally, such as the first time threshold is larger as the first time is longer, which can reduce the possibility that the first

task is overtime, such that the process of performing adjustment on the second task can converge.

[0056] At step 13, adjusting one or more time slices allocated to one or more second tasks in a target scheduling period according to a result of the determination, where the target scheduling period includes the current scheduling period or a next scheduling period of the current scheduling period.

[0057] In the embodiments of the present disclosure, the one or more time slices allocated to the one or more second tasks can be adjusted according to the result of determining whether the first task is overtime. The second task can be a task in the current scheduling period, or it can be a task in the next scheduling period of the current scheduling period.

[0058] When the second task indicates a task in the current scheduling period, the second task may include a task that is executed and completed in the current scheduling period, a task that not executed in the current scheduling period, and the first task. When the second task indicates a task in the next scheduling period of the current scheduling period, the second task may include all tasks in the next scheduling period.

[0059] For example, in a case that the first task in the current scheduling period is overtime, a length of a time slice allocated to a second task can be extended to obtain the extended time slice of the second task. In a case that all tasks in the current scheduling period are not overtime, the time slice allocated to the second task in the next scheduling period can be shortened to obtain the shortened time slice of the second task. If a task in the current scheduling period is overtime, on the basis of adjusting the length of the present scheduling period, the next scheduling period will inherit the adjusted length of the present scheduling period. In the scheduling method provided by the present disclosure, by continuously adjusting the scheduling period, such as extending the present scheduling period in a case of overtime (a length of the next scheduling period is equal to the extended length of the

present scheduling period) and reducing the next scheduling period in a case of not overtime, a relative "steady state" of the scheduling periods can be achieved, and at the same time, the time slice allocated to the task can be adjusted to a reasonable range.

[0060] Through the above scheduling scheme, the one or more time slices allocated to the one or more second tasks can be automatically adjusted, so that the one or more time slices allocated to the one or more second tasks in the target scheduling period meets requirements of QOS (Quality Of Service).

[0061] In a possible implementation, when it is determined that the first task is overtime, a length of the current scheduling period can be adjusted, and then based on the adjusted length of the current scheduling period, the one or more time slices allocated to the one or more second tasks in the current scheduling period are adjusted.

[0062] In this implementation, if it is determined that the first task is overtime, the length of the current scheduling period can be adjusted first, for example, the length of the current scheduling period can be extended by a fixed time period, or the length of the current scheduling period can be extended to several times the original. Then, according to the adjusted length of the current scheduling period, the length of the one or more time slices allocated to the one or more second tasks in the current scheduling period can be adjusted. For example, the one or more time slices allocated to each second task is extended by a fixed period length, or the one or more time slices allocated to each second task can be extended in a certain proportion. In this way, the length of the current scheduling period can be adjusted in a case of determining that the first task is overtime, and the one or more time slices of the one or more second tasks can be adjusted through the adjusted length of the current scheduling period.

[0063] In one example, for each of the one or more second tasks, adjusting a length of a time slice allocated to the second task in the current scheduling period based on the adjusted

length of the current scheduling period and a QOS (Quality Of Service) parameter of the second task so to achieve at least one of: maintaining the QOS parameter of the second task at a corresponding preset value, and maintaining a proportion of the time slice of the second task with respect to the current scheduling period at a corresponding preset value.

[0064] For example, there are three second tasks A, B, and C in the current scheduling period. Assuming that before an adjustment, a proportion of the second tasks A, B, and C in the current scheduling period is 3:3:4, that is the preset values are 3: 3: 4 respectively. After the adjustment, the proportion of the second tasks A, B and C with respect to the current scheduling period is still 3:3:4.

[0065] In this example, the QOS parameter can indicate a parameter that characterizes the QOS, and the QOS parameter can indicate a time slice share of a time slice corresponding to any one task in a scheduling period. The higher the QOS of a task, the more time slice shares the task occupies in the scheduling period. For example, the QOS of a task can correspond to a weight. The higher the QOS, the higher the weight. When calculating a time slice share, a sum of the weight of each task in the scheduling period can be calculated first, and then the weight of a task can be divided by the sum of the weight of each task, and the obtained value can be the time slice share of the time slice corresponding to the task with respect to the scheduling period. By allocating different QOS to different tasks, it can be ensured that tasks with high QOS can be processed first. A plurality of tasks can be included in a scheduling period, and each task can correspond to a QOS parameter. After adjusting the length of the current scheduling period, for each of the one or more second tasks, the length of the time slice allocated to the second task can be adjusted based on the adjusted length of the current scheduling period and the QOS parameter of the second task in the current scheduling period. In this way, at least one of the following can be achieved: the QOS parameter of the second task remains unchanged, and, a proportion of the time slice

of the second task with respect to the current scheduling period remains unchanged. Therefore, the length of the time slice of the second task can be adjusted according to the QOS, so as to meet requirements of QOS for each task.

[0066] In an example, in order to ensure that the QOS parameter of each task in the scheduling period remains unchanged after adjustment, the time period of the first task needs to be processed as well. The time period of the first task can be processed in the same way as other second tasks, for example, an allocated time slice of a first task is extended by a fixed time period length, or a time slice allocated to a first task can be extended in a certain proportion.

[0067] In an example, a number of overtime tasks that have been executed and completed in the current scheduling period can be obtained, and the executed and completed tasks can be referred to as the third tasks. An extension coefficient for extending the length of the current scheduling period can be determined according to the number of overtime third tasks, and the adjusted length of the current scheduling period can be acquired by extending the length of the current scheduling period based on the extension coefficient. The extension coefficient is positively correlated with the number of overtime third tasks.

[0068] In this example, the number of overtime third tasks in the current scheduling period can be counted. Every time a third task is executed and completed, if the third task is overtime, the number of overtime third tasks can be increased by one. The extension coefficient for extending the length of the current scheduling period can be determined according to the number of overtime third tasks, and the extension coefficient is positively correlated with the number of overtime third tasks. For example, the greater the number of overtime third tasks, the greater the extension coefficient. Then, the length of the current scheduling period can be extended based on the determined extension coefficient. For example, by multiplying the extension coefficient with the length of the current scheduling

period, the extended length of the current scheduling period can be obtained. The extension coefficient is greater than 1. In this way, the length of the current scheduling period can be reasonably extended through the number of overtime third tasks, so that the current scheduling period has a reasonable time period length.

[0069] In an example, the extension coefficient is determined based on an exponential function with the number of overtime third tasks as a variable.

[0070] In this example, the extension coefficient can be determined based on the exponential function with the number of overtime third tasks as a variable, such that the length of the current scheduling period can be extended by multiples through the extension coefficient to obtain the extended length of the current scheduling period. The extended length of the current scheduling period can be calculated according to the following formula (1).

$$T_{newtimeslice} = T_{oldtimeslice} * 2^{count} \quad \text{formula (1),}$$

where the $T_{newtimeslice}$ can indicate the extended length of the current scheduling period, the $T_{oldtimeslice}$ can indicate the length of the current scheduling period, the $count$ can indicate the counted time of overtime third tasks, and the 2^{count} can indicate the extension coefficient.

[0071] Adopting a binary exponential function as the extension coefficient, it can prevent the extended length of the previous scheduling period from increasing too fast, and it can also prevent the extended length of the current scheduling period from not converging due to subsequent slow growth.

[0072] For example, assuming that the current scheduling period includes 10 tasks, where the second, third, and fifth tasks are overtime. When the second task is overtime, since the first task is not overtime, the count indicates 1. According to the formula (1), $T_{newtimeslice}$ is twice of $T_{oldtimeslice}$, that is to say, compared with a length of the current scheduling

period before execution, this adjustment is to extend the length by twice. When the third task is overtime, the count indicates 2. According to the formula (1), $T_{newtimeslice}$ is 4 times of $T_{oldtimeslice}$. It should be noted that the $T_{oldtimeslice}$ is a length that has been extended once, that is, compared with the length of the current scheduling period before extending, this current adjustment is to extend to 4 times of the length which has been extended by 2 times. Similarly, when the fifth task is overtime, compared with the length of the current scheduling period before extending, this current adjustment is to extend 8 times of the length which has been extended by $2*4$ times.

[0073] In a possible implementation, a task processing time of a third task that has been executed and completed in the current scheduling period can be obtained, and then in response to that the task processing time of the third task is less than an adjusted length of the time slice of the third task, time compensation is performed on the third task in the current scheduling period. The second task includes the third task.

[0074] In this implementation, after adjusting the one or more time slices of the one or more second tasks in the current scheduling period, if the task processing time of any third task included in the one or more second tasks is less than the adjusted length of the time slice of the third task, the third task can be compensated in the current scheduling period.

[0075] Since the time slice of the third task that has been executed and completed cannot be directly extended, a way of performing time compensation on the third task can be used to keep the QOS parameter of each second task in the current scheduling period unchanged. Performing time compensation on the third task can be understood as allocating a corresponding time slice to the third task.

[0076] In an example of this implementation, a compensation time for the third task can be determined based on the task processing time of the third task and the adjusted length of

the time slice of the third task, and then based on the compensation time, a compensation task for the third task is added to a scheduling queue of the current scheduling period.

[0077] This example provides a way of performing time compensation on the third task. In this example, the compensation time for the third task can be determined according to the task processing time of the third task and the adjusted length of the time slice of the third task. For example, a second time difference between the adjusted length of the time slice of the third task and the task processing time of the third task can be calculated, and the second time difference can be used as the compensation time for the third task. Then, a task for compensating the third task can be added to the scheduling queue of the current scheduling period, and the calculated compensation time can be allocated to the third task when the compensation task is executing. In this way, time compensation on the executed and completed third task can be realized, so that the QOS parameter of the executed and completed third task remains unchanged.

[0078] In an example, based on the compensation time, the compensation task for the third task is added to an end of the scheduling queue of the current scheduling period.

[0079] In this example, the compensation task for the third task can be added to the end of the scheduling queue of the current scheduling period based on the compensation time for the third task, and a determined compensation time can be allocated to the compensation task, thereby realizing the time compensation for the third task. Adding the compensation task to the end of the scheduling queue of the current scheduling period will not disrupt an order of tasks in the scheduling queue of the current scheduling period, so that the tasks in the scheduling queue are scheduled in the original order.

[0080] In an example, the compensation time for the third task can be determined according to the QOS parameter of the executed and completed third task and the lengths before and after the adjustment of the current scheduling period.

[0081] In this example, a third time difference between the adjusted length of the current scheduling period and the length of the current scheduling period before the adjustment can be determined. Then, the compensation time for the third task can be determined according to the third time difference and the QOS parameter of the third task. For example, the compensation time for the third task can be obtained by multiplying the third time difference with the QOS parameter of the third task, such that the time compensation on the third task can be realized. The compensation time can be calculated according to the following formula (2).

$$t_{compensation} = (T_{newtimeslice} - T_{oldtimeslice}) * Share_{vGPU} \text{ formula (2),}$$

The $t_{compensation}$ can represent the compensation time, the $T_{newtimeslice}$ can indicate the adjusted length of the current scheduling period, the $T_{oldtimeslice}$ can indicate the length of the current scheduling period before the adjustment, and the $Share_{vGPU}$ can indicate the QOS parameter of the third task. The compensation time for each third task can be obtained by formula (2).

[0082] FIG. 2 shows an exemplary flowchart of extending one or more time slices allocated to a task according to embodiments of the present disclosure, which may include the following steps.

[0083] At step 201, determining whether a currently executing task is overtime.

[0084] In a case that the current task is overtime, S202 and subsequent steps can be executed. In a case that the current task is not overtime, S205 and subsequent steps can be executed.

[0085] At step 202, extending the length of the current scheduling period according to the number of overtime third tasks that have been executed and completed in the current scheduling period.

[0086] Optionally, the length of the current scheduling period can be adjusted according to a current cumulative number of overtime third tasks in the current scheduling period, that is, the cumulative number of overtime third tasks that have been executed and completed in the current scheduling period.

[0087] At step 203, adjusting one or more time slices allocated to each task in the current scheduling period according to the extended length of the current scheduling period.

[0088] At step 204, adding a compensation task for the executed and completed task to an end of a scheduling queue, and allocating corresponding compensation time to the compensation task.

[0089] By extending a time slice of a task, the one or more time slices allocated to each task in the current scheduling period can be extended when the currently executing task is overtime, so that the extended time slice can be adapted to satisfy the corresponding QOS.

[0090] At step 205, determining whether the current scheduling period is ended.

[0091] In a case that the current scheduling period is not ended, step S207 is executed. In a case that the current scheduling period is ended, step S206 is executed.

[0092] At step 206, terminating the current process, and inheriting the length of the current scheduling period to the next scheduling period.

[0093] At step 207, scheduling the next task to execute the task, updating the next task to the currently executing task, and return to step S201.

[0094] In a possible implementation, a length of the next scheduling period of the current scheduling period can be adjusted in a case of determining that each task in the current scheduling period is not overtime, and then the one or more time slices allocated to the one or more second tasks in the next scheduling period are determined based on the adjusted length of the next scheduling period.

[0095] In this example, when each task in the current scheduling period is not overtime, the length of the next scheduling period of the current scheduling period can be reduced to obtain a shortened length of the next scheduling period. For example, a certain time period can be reduced on the basis of the length of the next scheduling period, or the length of the next scheduling period can be multiplied with a certain coefficient to obtain the shortened length of the next scheduling period. Then, the lengths of the one or more time slices allocated to each second task in the next scheduling period can be determined according to the adjusted length of the next scheduling period. For example, the length of the time slice allocated to each second task in the next scheduling period may be reduced, so that a sum of the reduced lengths of the one or more time slices is equal to the adjusted length of the next scheduling period. According to the reduced length of the time slice of each second task, the time slice allocated to each second task can be further determined, such that the time slice of the second task in the next scheduling period can be automatically reduced when the time slice allocated to the task in the current scheduling period is too long, such that resources of the virtual image processing unit can be effectively used.

[0096] In an example, the length of the next scheduling period can be reduced based on a preset reduction coefficient.

[0097] In this example, the reduction coefficient can be set according to actual application scenarios, and can be set to a real number less than 1, for example, the reduction coefficient is set to $1/2$. When reducing the length of the next scheduling period based on the reduction coefficient, the length of the next scheduling period can be multiplied with the reduction coefficient to obtain the reduced length of the next scheduling period. By using an appropriate reduction coefficient to reduce the length of the next scheduling period, it is possible to avoid the reduced length of the next scheduling period from being too small due to an excessive reduction.

[0098] In an example, for each of the one or more second tasks, a length of a time slice allocated to the second task in the next scheduling period can be determined based on the adjusted length of the next scheduling period and a QOS parameter corresponding to the second task, so as to achieve at least one of: the QOS parameter of the second task is maintained at a corresponding preset value, and a proportion of the time slice of the second task with respect to the next scheduling period is maintained at a corresponding preset value.

[0099] In this example, after adjusting the length of the next scheduling period, for each of the one or more second tasks, the length of the time slice allocated to the second task can be determined according to the QOS parameter corresponding to the second task in the next scheduling period and the adjusted length of the next scheduling period. For example, for each of the one or more second tasks, the QOS parameter corresponding to the second task is multiplied with the adjusted length of the next scheduling period to obtain the length of the time slice allocated to the second task, and then the time slice allocated to the second task is determined. In this way, the QOS parameter of each second task can be kept unchanged, and/or the proportion of the time slice of each second task with respect to the current scheduling period remains unchanged, thereby meeting the requirement of QOS.

[0100] FIG. 3 shows an exemplary flowchart of reducing one or more time slices allocated to a task according to embodiments of the present disclosure, which may include the following steps.

[0101] At step 301, determining whether each task in the current scheduling period is overtime.

[0102] At step 302, reducing, in a case that each task in the current scheduling period is not overtime, a length of the next scheduling period based on a reduction coefficient to obtain the reduced length of the next scheduling period.

[0103] At step 303, reducing the time slice allocated to each second task in the next scheduling period based on the reduced length of the next scheduling period.

[0104] At step 304, scheduling tasks in the next scheduling period.

[0105] By reducing a time slice of a task, the one or more time slices allocated to each task in the next scheduling period can be automatically reduced when each task in the current scheduling period is not overtime, so that the reduced time slice can be adapted to satisfy the corresponding QOS.

[0106] In a possible implementation, respective historical task processing time of each of historical tasks executed and completed in a preset time period can be obtained, and an initial length of the current scheduling period can be determined according to statistical information of the respective historical task processing time.

[0107] In this implementation, the initial length of the current scheduling period can be determined through the historical task processing time in the preset time period. The preset time period can be set according to actual application scenarios. For example, the historical task processing time of a plurality of historical tasks executed and completed in the last 24 hours can be obtained. The statistical information of the historical task processing time may include information of a historical task, a cumulative frequency of a historical task, etc., and then the initial length of the current scheduling period can be determined according to the statistical information of the historical task processing time. For example, a curve showing the relationship between the historical task processing time and the cumulative frequency of the historical task can be determined according to historical task processing time of a plurality of historical tasks executed and completed, and then an appropriate historical task processing time can be selected as the initial length of the current scheduling period based on the curve.

[0108] FIG. 4 illustrates a curve showing the relationship between historical task processing time and cumulative frequency according to embodiments of the present disclosure. In the figure, the abscissa corresponds to the historical task processing time, the unit of the abscissa is ms, and the ordinate corresponds to the cumulative frequency. As shown in FIG 4, the cumulative frequency of 70% corresponds to about 100 ms, which can be understood that in the preset time period, the task processing time of 70% of the tasks is within 100 ms. For example, the historical task processing time corresponding to a cumulative frequency of 70% can be determined as the initial length of the current scheduling period. Alternatively, the historical task processing time corresponding to the cumulative frequency of 70% can be multiplied by a multiple, for example, the historical task processing time corresponding to the cumulative frequency of 70% can be multiplied by 10 to obtain the initial length of the current scheduling period.

[0109] In an example of this implementation, for each task in the current scheduling period, a time period of an initial time slice allocated to the task can be determined according to the initial length of the current scheduling period and the QOS parameter corresponding to the task in the current scheduling period.

[0110] In this example, the initial time period allocated to each task can be calculated according to the QOS parameter corresponding to each task in the current scheduling period and the initial length of the current scheduling period. For example, the current scheduling period includes 3 tasks, task A, task B and task C, where the time slice share corresponding to the task A is 3, the time slice share corresponding to the task B is 3, and the time slice share corresponding to the task C is 4. Then the initial time slice of the task A can be 0.3 initial time period, the initial time slice of the task B can be 0.3 initial time period, and the initial time slice of the task C can be 0.4 initial time period. In this way, the initial time slice

of each task in a scheduling period can be allocated according to corresponding QOS parameter, and the requirements of QOS are satisfied.

[0111] In a possible implementation, the initial length of the current scheduling period can also be set according to actual application scenarios, for example, the initial length is set to 100 ms. Then the initial time slice allocated to each task in the current scheduling period can be determined based on the initial length of the current scheduling period and the QOS parameter corresponding to each task in the current scheduling period.

[0112] In the scheduling scheme provided by the embodiments of the present disclosure, the time slice allocated to the task can be automatically adjusted when the length of the time slice allocated to the task is too long or too short, so as to realize the effect of reasonable configuring the time slice of the task.

[0113] It can be understood that the various method embodiments mentioned in this disclosure can be combined with each other to form a combined embodiment without violating the principle and logic, which will not be repeated in this disclosure due to space limitations.

[0114] In addition, the present disclosure also provides an apparatus, an electronic device, a computer-readable storage medium, and a program, all of which can be used to implement any scheduling method provided in the present disclosure. For corresponding technical solutions and descriptions, please refer to the corresponding records in the method section, which will not be repeated.

[0115] Those skilled in the art can understand that in the above method of specific implementations, the writing order of each step does not mean a strict execution order and constitutes any limitation on the implementation process, the specific execution order of each step should be determined based on its function and the possible internal logic.

[0116] FIG. 5 shows a block diagram of a scheduling apparatus according to an embodiment of the present disclosure. As shown in FIG. 5, the apparatus includes an obtaining unit 51, a determining unit 52, and an adjustment unit 53.

[0117] The obtaining unit 51, configured to obtain a task processing time of a first task in a current scheduling period.

[0118] The determining unit 52, configured to determine whether the first task is overtime based on the task processing time and a first time slice allocated to the first task in the current scheduling period.

[0119] The adjusting unit 53, configured to adjust one or more time slices allocated to one or more second tasks in a target scheduling period according to a result of the determination, where the target scheduling period includes the current scheduling period or a next scheduling period of the current scheduling period.

[0120] In one or more alternative embodiments, the determining unit 52 is specifically configured to determine a first time difference between a deadline of the task processing time and a deadline of the first time slice; and in a case that the first time difference is greater than a first time threshold, determine that the first task is overtime.

[0121] In one or more alternative embodiments, the first time threshold is determined based on a length of the first time slice.

[0122] In one or more alternative embodiments, the adjusting unit 53 is specifically configured to, in a case of determining that the first task is overtime, adjust a length of the current scheduling period; and adjust the one or more time slices allocated to the one or more second tasks in the current scheduling period based on the adjusted length of the current scheduling period.

[0123] In one or more alternative embodiments, the adjusting unit 53 is specifically configured to, for each of the one or more second tasks, adjust a length of a time slice

allocated to the second task in the current scheduling period based on the adjusted length of the current scheduling period and a Quality Of Service QOS parameter of the second task to achieve at least one of: maintain the QOS parameter of the second task at a corresponding preset value, and maintain a proportion of the time slice of the second task with respect to the current scheduling period at a corresponding preset value.

[0124] In one or more alternative embodiments, the adjusting unit 53 is specifically configured to, obtain a number of overtime third tasks in the current scheduling period, where the third tasks indicate one or more tasks that have been executed and completed in the current scheduling period; determine an extension coefficient for extending the length of the current scheduling period according to the number of overtime third tasks, where the extension coefficient is positively correlated with the number of overtime third tasks; and acquire the adjusted length of the current scheduling period by extending the length of the current scheduling period based on the extension coefficient.

[0125] In one or more alternative embodiments, the extension coefficient is determined based on an exponential function with the number of overtime third tasks as a variable.

[0126] In one or more alternative embodiments, the adjusting unit 53 is further configured to, obtain a task processing time of a third task in the current scheduling period, where the one or more second tasks include the third task; and in response to that the task processing time of the third task is less than an adjusted length of a time slice of the third task, perform time compensation on the third task in the current scheduling period.

[0127] In one or more alternative embodiments, the adjusting unit 53 is specifically configured to, determine a compensation time for the third task based on the task processing time of the third task and the adjusted length of the time slice of the third task; and add, based on the compensation time, a compensation task for the third task to a scheduling queue of the current scheduling period.

[0128] In one or more alternative embodiments, the adjusting unit 53 is specifically configured to add, based on the compensation time, the compensation task for the third task to an end of the scheduling queue of the current scheduling period.

[0129] In one or more alternative embodiments, the adjusting unit 53 is specifically configured to, in a case of determining that each task in the current scheduling period is not overtime, adjust a length of the next scheduling period of the current scheduling period; and determine the one or more time slices allocated to the one or more second tasks in the next scheduling period based on the adjusted length of the next scheduling period.

[0130] In one or more alternative embodiments, the adjusting unit 53 is specifically configured to reduce the length of the next scheduling period based on a preset reduction coefficient.

[0131] In one or more alternative embodiments, the adjusting unit 53 is specifically configured to, for each of the one or more second tasks, determine a length of a time slice allocated to the second task in the next scheduling period based on the adjusted length of the next scheduling period and a QOS parameter of the second task to achieve at least one of: maintain the QOS parameter of the second task at a corresponding preset value, and maintain a proportion of the time slice of the second task with respect to the next scheduling period at a corresponding preset value.

[0132] In one or more alternative embodiments, the apparatus further includes: an initial period determining unit, configured to obtain respective historical task processing time of each of historical tasks executed and completed by a virtual image processing unit in a preset time period; and determine an initial length of the current scheduling period according to statistical information of the respective historical task processing time.

[0133] In some embodiments, functions or modules included in the apparatus provided in the embodiments of the present disclosure can be used to execute the methods described in

the above method embodiments. For specific implementation, please refer to the description of the above method embodiments, which will not be repeated here for brevity.

[0134] An embodiment of the present disclosure also provides an electronic device, including: a processor; a memory for storing instructions executed by the processor; where the processor is used to call the instructions stored in the memory, to implement the above scheduling method.

[0135] The electronic device can be provided as a terminal, a server or a device of other form.

[0136] FIG. 6 is a block diagram illustrating an electronic device 600 according to an exemplary embodiment. For example, the electronic device 600 can be provided as a server. Referring to FIG. 6, an electronic device 600 may include a processing component 622 which further includes one or more processors, and memory resource represented by a memory 632 which is used for storing instructions executable by the processing component 622, for example the instructions may indicate application programs. The application program stored in the memory 632 may include one or more modules, each of the one or more modules corresponds to a set of instructions. In addition, the processing component 622 may be configured to execute instructions to implement the above-described scheduling methods.

[0137] The electronic device 600 may also include a power component 626 which is configured to perform power management on the electronic device 600, a wired or wireless network interface 650 configured to connect the electronic device 600 to a network, and an input/output (I/O) interface 658. The electronic device 600 can operate an operating system stored in the memory 632, such as Windows ServerTM, Mac OS XTM, UnixTM, LinuxTM, FreeBSDTM or the like.

[0138] In an exemplary embodiment, a non-transitory computer readable storage medium is also provided, such as a memory 632 including computer program instructions, the computer program instructions may be executed by the processing component 622 of the electronic device 600 to implement the above-described scheduling methods.

[0139] The present disclosure can be provided as a method, a system, and /or a computer program product. The computer program product may include a computer-readable storage medium loaded with computer-readable program instructions for causing a processor to implement various aspects of the present disclosure.

[0140] The computer-readable storage medium may be a tangible device that can hold and store instructions used by an instruction execution device. The computer-readable storage medium may be, for example, but not limited to, an electrical storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. More specific examples of the computer-readable storage media (non-exhaustive list) include: portable computer disks, hard disks, random access memory (RAM), read-only memory (ROM), erasable programmable read-only memory (EPROM or flash memory), static random access memory (SRAM), portable compact disk read-only memory (CD-ROM), digital versatile disk (DVD), memory stick, floppy disk, mechanical encoding device, such as punch cards with instructions stored thereon, protruding structure in the groove, and any suitable combination of the foregoing. The computer-readable storage medium used here is not interpreted as a transient signal itself, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating via waveguides or other transmission media (for example, light pulses propagate via fiber optic cables), or electrical signals transmitted via wires.

[0141] The computer-readable program instructions described herein can be downloaded from a computer-readable storage medium to various computing/processing devices, or to an external computer or an external storage device via a network, such as the Internet, a local area network, a wide area network, and/or a wireless network. The network may include copper transmission cables, optical fiber transmission, wireless transmission, routers, firewalls, switches, gateway computers, and/or edge servers. The network adapter card or network interface in each computing/processing device receives computer-readable program instructions from the network, and forwards the computer-readable program instructions so that the computer-readable program instructions may be stored in the computer-readable storage medium of each computing/processing device.

[0142] The computer program instructions used to perform the operations of the present disclosure may be assembly instructions, instruction set architecture (ISA) instructions, machine instructions, machine-related instructions, microcode, firmware instructions, state setting data, or source code or object code written in one or more programming languages or any combination of thereof, the programming languages including object-oriented programming languages such as Smalltalk, C++, etc., and conventional procedural programming languages such as "C" language or similar programming languages. Computer-readable program instructions can be executed entirely on a user's computer, partly on the user's computer, executed as a stand-alone software package, partly executed on the user's computer and partly executed on a remote computer, or executed entirely on the remote computer or server. In a situation involving a remote computer, the remote computer can be connected to a user's computer through any kind of network, including a local area network (LAN) or a wide area network (WAN), or it can be connected to an external computer (for example, using an Internet service provider to connect to an external computer via the Internet). In some embodiments, an electronic circuit, such as a

programmable logic circuit, a field programmable gate array (FPGA), or a programmable logic array (PLA), can be personalized customized by using status information of the computer-readable program instructions. The computer-readable program instructions are executed by the electronic circuit to realize various aspects of the present disclosure.

[0143] Various aspects of the present disclosure are described herein with reference to flowcharts and/or block diagrams of the methods, devices (systems) and computer program products in embodiments of the present disclosure. It should be understood that each block of the flowcharts and/or block diagrams, and combinations of blocks in the flowcharts and/or block diagrams can be implemented by computer-readable program instructions.

[0144] These computer-readable program instructions can be provided to processors of general-purpose computers, special-purpose computers, or other programmable data processing devices to produce a machine that produces a device that implements the functions/actions specified in one or more blocks in the flowcharts and/or block diagrams, when these instructions are executed by a processor of a computer or other programmable data processing device. It is also possible to store these computer-readable program instructions in a computer-readable storage medium. These instructions make computers, programmable data processing apparatuses, and/or other devices work in a specific manner. Thus, the computer-readable medium storing the instructions includes a manufacture, which includes instructions for implementing various aspects of the functions/actions specified in one or more blocks in the flowcharts and/or block diagrams.

[0145] It is also possible to load the computer-readable program instructions onto a computer, other programmable data processing device, or other equipment, so that a series of operation steps are executed on the computer, other programmable data processing device, or other equipment to generate a computer-implemented process, such that the instructions executed on the computer, other programmable data processing apparatus, or other

equipment can implement the functions/actions specified in one or more blocks in the flowcharts and/or block diagrams.

[0146] The flowcharts and block diagrams in the drawings show architectures, functions, and operations that may be implemented according to the system, the method, and the computer program product described in a plurality of embodiments of the present disclosure. In this regards, each block in the flowchart or block diagram may represent a module, a program segment, or a part of an instruction, and the module, the program segment, or the part of the instruction contains one or more executable instructions for realizing a specified logic function. In some alternative implementations, the functions marked in a block may also occur in an order which is different from the order marked in the drawings. For example, two consecutive blocks can actually be executed in parallel, or they can sometimes be executed in the reverse order, depending on the functions involved. It should also be noted that each block in the block diagram and/or flowchart, and a combination of the blocks in the block diagram and/or flowchart, can be implemented by a dedicated hardware-based system that performs specified functions or actions, or it can be realized by a combination of a dedicated hardware and computer instructions.

[0147] The embodiments of the present disclosure have been described above, and the above description is exemplary, not exhaustive, and is not limited to the disclosed embodiments. Without departing from the scope and spirit of the described embodiments, many modifications and changes are obvious to those of ordinary skill in the art. The choice of terms used herein is intended to best explain the principles, practical applications, or technical improvements of the technologies in the market, or to enable other ordinary skilled in the art to understand the embodiments disclosed herein.

CLAIMS

1. A method of scheduling, comprising:

obtaining a task processing time of a first task in a current scheduling period;

determining whether the first task is overtime based on the task processing time and a first time slice allocated to the first task in the current scheduling period; and

adjusting one or more time slices allocated to one or more second tasks in a target scheduling period according to a result of the determination, wherein the target scheduling period comprises the current scheduling period or a next scheduling period of the current scheduling period.

2. The method according to claim 1, wherein, determining whether the first task is overtime based on the task processing time and the first time slice allocated to the first task in the current scheduling period, comprises:

determining a first time difference between a deadline of the task processing time and a deadline of the first time slice; and

determining, in a case that the first time difference is greater than a first time threshold, that the first task is overtime.

3. The method according to claim 2, wherein the first time threshold is determined based on a length of the first time slice.

4. The method according to any one of claims 1 to 3, wherein adjusting the one or more time slices allocated to the one or more second tasks in the target scheduling period according to a result of the determination, comprises:

adjusting, in a case of determining that the first task is overtime, a length of the current scheduling period; and

adjusting the one or more time slices allocated to the one or more second tasks in the current scheduling period based on the adjusted length of the current scheduling period.

5. The method according to claim 4, wherein adjusting the one or more time slices allocated to the one or more second tasks in the current scheduling period based on the adjusted length of the current scheduling period, comprises:

for each of the one or more second tasks,

adjusting a length of a time slice allocated to the second task in the current scheduling period based on the adjusted length of the current scheduling period and a Quality Of Service QOS parameter of the second task to achieve at least one of:

maintaining the QOS parameter of the second task at a corresponding preset value, and

maintaining a proportion of the time slice of the second task with respect to the current scheduling period at a corresponding preset value.

6. The method according to claim 4 or 5, wherein, adjusting, in a case of determining that the first task is overtime, the length of the current scheduling period, comprises:

obtaining a number of overtime third tasks in the current scheduling period, wherein the third tasks indicate one or more tasks that have been executed and completed in the current scheduling period;

determining an extension coefficient for extending the length of the current scheduling period according to the number of overtime third tasks, wherein the extension coefficient is positively correlated with the number of overtime third tasks; and

acquiring the adjusted length of the current scheduling period by extending the length of the current scheduling period based on the extension coefficient.

7. The method according to claim 6, wherein, the extension coefficient is determined based on an exponential function with the number of overtime third tasks as a variable.

8. The method according to any one of claims 4 to 7, further comprises:

obtaining a task processing time of a third task in the current scheduling period, wherein the one or more second tasks comprise the third task; and

performing, in response to that the task processing time of the third task is less than an adjusted length of a time slice of the third task, time compensation on the third task in the current scheduling period.

9. The method according to claim 8, wherein performing, in response to that the task processing time of the third task is less than the adjusted length of the time slice of the third task, time compensation on the third task in the current scheduling period, comprises:

determining a compensation time for the third task based on the task processing time of the third task and the adjusted length of the time slice of the third task; and

adding, based on the compensation time, a compensation task for the third task to a scheduling queue of the current scheduling period.

10. The method according to claim 9, wherein adding, based on the compensation time, a compensation task for the third task to the scheduling queue of the current scheduling period, comprises:

adding, based on the compensation time, the compensation task for the third task to an end of the scheduling queue of the current scheduling period.

11. The method according to any one of claims 1 to 3, wherein adjusting the one or more time slices allocated to the one or more second tasks in the target scheduling period according to a result of the determination, comprises:

adjusting, in a case of determining that each task in the current scheduling period is not overtime, a length of the next scheduling period of the current scheduling period; and

determining the one or more time slices allocated to the one or more second tasks in the next scheduling period based on the adjusted length of the next scheduling period.

12. The method according to claim 11, wherein adjusting the length of the next scheduling period of the current scheduling period, comprises:

reducing the length of the next scheduling period based on a preset reduction coefficient.

13. The method according to claim 11 or 12, wherein determining the one or more time slices allocated to the one or more second tasks in the next scheduling period based on the adjusted length of the next scheduling period, comprises:

for each of the one or more second tasks,

determining a length of a time slice allocated to the second task in the next scheduling period based on the adjusted length of the next scheduling period and a QOS parameter of the second task to achieve at least one of:

maintaining the QOS parameter of the second task at a corresponding preset value, and

maintaining a proportion of the time slice of the second task with respect to the next scheduling period at a corresponding preset value.

14. The method according to any one of claims 1 to 13, further comprises:

obtaining respective historical task processing time of each of historical tasks executed and completed in a preset time period;

determining an initial length of the current scheduling period according to statistical information of the respective historical task processing time.

15. An apparatus for scheduling, comprising:

an obtaining unit, configured to obtain a task processing time of a first task in a current scheduling period;

a determining unit, configured to determine whether the first task is overtime based on the task processing time and a first time slice allocated to the first task in the current scheduling period; and

an adjusting unit, configured to adjust one or more time slices allocated to one or more second tasks in a target scheduling period according to a result of the determination, wherein the target scheduling period comprises the current scheduling period or a next scheduling period of the current scheduling period.

16. The apparatus according to claim 15, wherein the adjusting unit is further configured to, obtain a task processing time of a third task in the current scheduling period, wherein the one or more second tasks comprise the third task; and in response to that the task processing time of the third task is less than an adjusted length of a time slice of the third task, perform time compensation on the third task in the current scheduling period.

17. The apparatus according to claim 16, wherein the adjusting unit is specifically configured to, determine a compensation time for the third task based on the task processing time of the third task and the adjusted length of the time slice of the third task; and add, based on the compensation time, a compensation task for the third task to a scheduling queue of the current scheduling period.

18. The apparatus according to claim 17, wherein the adjusting unit is specifically configured to add, based on the compensation time, the compensation task for the third task to an end of the scheduling queue of the current scheduling period.

19. An electronic device, comprising:

a processor;

a memory for storing instructions executed by the processor;

wherein, the processor is used to call the instructions stored in the memory, to implement the method according to any one of claims 1 to 14.

20. A computer readable storage medium, having computer program instructions stored thereon, wherein in a case that the computer program instructions are executed by a processor, the method according to any one of claims 1 to 14 is implemented.