

METHOD FOR RESOURCE ALLOCATION DURING DIALOG AUDIO DATA PROCESSING, DEVICE, ELECTRONIC DEVICE AND STORAGE MEDIUM

FIELD OF INVENTION

5

The present invention relates to a method for resource allocation during dialog audio data processing, and related devices, and storage medium.

BACKGROUND

10

In the field of intelligent voice services, line resources, freeswitch resources, and server resources typically exhibit relative scarcity during specific periods, particularly during peak business hours, when these resources are often fully utilized; whereas during off-peak hours, the resources often go unused. Therefore, to optimize task processing efficiency and increase the number of tasks completed within a limited time, an efficient and rapid resource allocation mechanism is crucial.

15

Existing resource allocation strategies include, for example, round-robin allocation, random allocation, and minimum load allocation. Round-robin allocation involves sequentially allocating available resources to pending tasks according to a predefined order. Random allocation involves irregularly assigning tasks to any available resource. Minimum load allocation dynamically assigns tasks to the resource with the lowest current load based on real-time load conditions.

20

However, the aforementioned allocation strategies exhibit slower allocation speeds when facing large-scale task allocation, which to some extent limits the overall processing capability. Additionally, due to the relatively scattered task allocation, the impact range can be relatively large when a certain resource node fails, potentially leading to impediment in the execution of multiple tasks and correspondingly reducing system stability and reliability. Therefore, it is important to improve resource allocation strategies to enhance allocation efficiency and reduce the impact of single-point failures on the entire system.

25

30

SUMMARY

In a first aspect, there is provided a method for necessary resource allocation during dialog audio data processing, characterized in that the method comprises the following steps:

obtaining the remaining demand for necessary resources required by the target process and multiple necessary resource sets associated with said process;

generating the total available resources of said multiple necessary resource sets based on the available resource remainder of each set;

when the remaining demand for necessary resources required by the target process is less than or equal to the total available resources of said multiple necessary resource sets, cyclically executing the following steps until the remaining demand for necessary resources reaches zero:

selecting the necessary resource set with the largest available resource remainder from said multiple necessary resource sets as the first set;

if the available resource remainder of the first set is greater than or equal to the remaining demand for necessary resources required by the target process, allocating the target process to use said first set, and deducting the available resource remainder of said first set based on the remaining demand for necessary resources required by the target process demand, and deducting the remaining demand for necessary resources required by said target process to zero;

if the available resource remainder of the first set is less than the remaining demand for necessary resources required by the target process, allocating the entire available resource remainder of the first set to the target process, and deducting the remaining demand for necessary resources required by said target process based on the available resource remainder, and deducting the available resource remainder of said first set to zero.

In a second aspect, there is provided a device for necessary resource allocation during dialog audio data processing, characterized in that the device comprises an allocation preparation module, a total available resource management module, and a necessary resource allocation execution module, whereby

the allocation preparation module is configured to obtain the remaining demand for necessary resources required by the target process and multiple necessary resource sets associated with said process;

the total available resource management module is configured to generate the total
5 available resources of said multiple necessary resource sets based on the available resource remainder of each set;

the necessary resource allocation execution module is configured to, when the remaining demand for necessary resources required by the target process is less than or equal to the total available resources of said multiple necessary resource sets,
10 cyclically execute the following steps until the remaining demand for necessary resources reaches zero:

selecting the necessary resource set with the largest available resource remainder from said multiple necessary resource sets as the first set;

if the available resource remainder of the first set is greater than or equal to the
15 remaining demand for necessary resources required by the target process, allocating the target process to use said first set, and deducting the available resource remainder of said first set based on the remaining demand for necessary resources required by the target process demand, and deducting the remaining demand for necessary resources required by said target process to zero;

if the available resource remainder of the first set is less than the remaining demand
20 for necessary resources required by the target process, allocating the entire available resource remainder of the first set to the target process, and deducting the remaining demand for necessary resources required by said target process based on the available resource remainder, and deducting the available resource remainder of said
25 first set to zero.

Analogous aspects relating to a computer-readable storage medium and an electronic device are also provided.

30 It will be appreciated that the broad forms of the invention and their respective features can be used in conjunction, interchangeably and/or independently, and reference to separate broad forms is not intended to be limiting.

DESCRIPTION OF FIGURES

By reading the detailed description of exemplary embodiments in the following text, those skilled in the art will understand the advantages and benefits described herein, as well as other advantages and benefits. The accompanying drawings are only for the purpose of demonstrating exemplary embodiments and are not considered a limitation on the application. And throughout all drawings, the same components are represented by the same numbers. In the attached drawings:

FIG 1 illustrates the system architecture of the necessary resource allocation system for dialog audio data processing.

FIG 2A illustrates a flowchart illustrating the failure of necessary resource allocation.

FIG 2B illustrates a flowchart illustrating the process of necessary resource allocation.

FIG 3 illustrates the sorting of necessary resource sets.

FIG 4 illustrates the resorting of necessary resource sets after the initial allocation.

FIG 5 illustrates the sorting of necessary resource sets based on the quantity of resources occupied.

FIG 6 illustrates an exemplary flowchart of the necessary resource allocation method for dialog audio data processing according to an embodiment of the application.

FIG 7 illustrates another exemplary flowchart of the necessary resource allocation method for dialog audio data processing according to an embodiment of the application.

FIG 8 illustrates another exemplary flowchart of the necessary resource allocation method for dialog audio data processing according to an embodiment of the application.

FIG 9 illustrates another exemplary flowchart of the necessary resource allocation method for dialog audio data processing according to an embodiment of the application.

FIG 10 illustrates another exemplary flowchart of the necessary resource allocation method for dialog audio data processing according to an embodiment of the application.

FIG 11 illustrates an exemplary structural diagram of the necessary resource allocation device for dialog audio data processing according to an embodiment of the application.

FIG 12 shows an exemplary structural diagram of an electronic device capable of implementing the method according to an embodiment of the application.

DETAILED DESCRIPTION

5

Exemplary embodiments of the present application will be described in more detail below with reference to all accompanying drawings. Although the exemplary embodiments of the present application are shown in the accompanying drawings, it should be understood that the present application can be implemented in various forms and should not be limited by the embodiments described here. On the contrary, these embodiments are provided to enable a more thorough understanding of the present application and to fully convey the scope of the present application to those skilled in the art.

10 In the description of the embodiments of the application, it should be understood that terms such as "including" or "having" are intended to indicate the presence of disclosed features, numbers, steps, actions, components, parts, or combinations thereof in this specification, and do not exclude the possibility of the existence of one or more other features, numbers, steps, actions, components, parts, or combinations thereof.

20 Unless otherwise specified, "/" represents the meaning of "or". For example, A/B can represent A or B; "and/or" in this document is just a way to describe the relationship between associated objects, indicating that there can be three types of relationships. For example, A and/or B can represent three situations: the existence of A alone, the simultaneous existence of A and B, and the existence of B alone.

30 Terms such as "first", "second", and the like are used to distinguish between identical or similar technical features for descriptive convenience only and should not be interpreted as indicating or implying the relative importance or quantity of these technical features. Thus, features defined by "first", "second", etc., can explicitly or implicitly include one or more of these features. In the description of the embodiments of the application, unless otherwise specified, the term "multiple" means two or more.

It should also be noted that, without conflict, the embodiments and features in the embodiments in the application can be combined with each other. The following will refer to the accompanying drawings and combine embodiments to illustrate the application in detail.

5

In an embodiment of the application, "freeswitch" is a cross-platform open-source telephony switch platform with strong scalability, capable of providing routing and interconnecting communication protocols for audio, video, text, or any other form of media.

10

In intelligent voice services, necessary resources such as lines, freeswitch, and servers typically all reach full capacity during peak hours, while all resources are idle during off-peak hours. However, existing resource allocation methods are slow, affecting business efficiency. Additionally, due to the dispersed allocation, the impact range is larger when resources fail.

15

As shown in FIG 1, an embodiment of the application provides a resource allocation system 100 configured for dialog audio data processing, comprising a backend allocation component 110, an allocation server 120, a dialog audio data processing execution server 130, a necessary resource repository 140, and a voice service request component 150. Configuration personnel operate the backend allocation component 110 to send an allocation request to the allocation server 120, which comprises information such as, for example, a name of the target process, a remaining demand for necessary resources, and types of necessary resources. The allocation server 120 is configured to retrieve corresponding information about multiple necessary resource sets from the necessary resource repository 140, such as the available resource remainder of each necessary resource set, and then allocates resources based on the available resource remainder and the required resource quantity to generate allocation configuration information. The allocation server 120 is also configured to send the allocation configuration information to the dialog audio data processing execution server 130 and the necessary resource repository 140 respectively, and send allocation success information to the backend allocation

20

25

30

component 110. When executing the target process, the voice service request component 150 is configured to initiate a voice request to the dialog audio data processing execution server 130, which is configured to invoke necessary resources from the necessary resource repository 140 according to the allocation configuration information to provide services for the dialog audio data processing process to users.

In an embodiment of the application, the process of the allocation server 120 generating allocation configuration information is as follows:

- obtaining the remaining demand for necessary resources required by the target process and multiple necessary resource sets associated with the target process;
- generating the total available resources of multiple necessary resource sets based on the available resource remainder of each necessary resource set.

When the remaining demand for necessary resources required by the target process is less than or equal to the total available resources of multiple necessary resource sets, cyclically executing the following steps until the remaining demand for necessary resources required by the target process is zero:

- selecting the necessary resource set with the largest available resource remainder from the multiple necessary resource sets as the first set;
- if the available resource remainder of the first set is greater than or equal to the remaining demand for necessary resources required by the target process, allocating the target process to use the first set, and deducting the available resource remainder of said first set based on the remaining demand for necessary resources required by the target process demand, and deducting the remaining demand for necessary resources required by said target process to zero;

if the available resource remainder of the first set is less than the remaining demand for necessary resources required by the target process, allocating the entire available resource remainder of the first set to the target process, and deducting the remaining demand for necessary resources required by said target process based on the

available resource remainder, and deducting the available resource remainder of said first set to zero remainder.

As shown in FIG 2A, multiple necessary resource sets include necessary resource set A, necessary resource set B, and necessary resource set C, with available resource remainders of 10, 20, and 30, respectively, resulting in a total available resources of 60. When the remaining demand for necessary resources required by the target process is 70, the allocation fails because the remaining demand exceeds the total available resources.

As shown in FIG 2B, when the remaining demand for necessary resources required by the target process is 40, the allocation proceeds since the remaining demand is less than the total available resources. The necessary resource set C with the largest available resource remainder is preferentially allocated, but its available resource remainder of 30 is less than the required quantity, so all resources in set C are allocated to the target process, reducing the remaining demand to 10. Then, the necessary resource set B with the next largest available resource remainder is allocated, satisfying the remaining demand, and the allocation ends.

The allocation server 120 is configured to generate allocation configuration information based on the relationship between necessary resource sets and the target process, which is then sent to the dialog audio data processing execution server 130 and the necessary resource repository 140. The dialog audio data processing execution server 130 is configured to invoke resources according to the configuration information for use.

The allocation system described in an embodiment of the application operates based on the remaining necessary resource demand required by the target process. It assesses whether the total available resources across multiple necessary resource sets can meet this demand. Once confirmed, the system identifies the necessary resource set with the greatest available resource remainder for allocation. If the available resources within a single set are sufficient to fulfill the target process's necessary resource requirements, all resources from that set are assigned to the target process. Alternatively, if the set with the largest available balance cannot fully

satisfy the target process's demand, all its resources are still allocated to the target process. This cycle continues, with subsequent allocations drawn from the remaining sets with the highest available resource remainder, until the entire allocation process is complete. This method ensures efficient resource allocation with minimal iterations, centralized task assignment, and reduced impact on the dialog audio data processing in case of resource failures, thereby enhancing switching efficiency.

In an embodiment of the application, before generating the total available resources of multiple necessary resource sets based on the available resource remainder of each necessary resource set, the allocation server 120 is also configured to perform the following operation:

- multiple necessary resource sets are sorted in descending order based on their available resource remainders. This facilitates finding the resource set with the largest available resource remainder. After finding the resource set with the largest available resource remainder, it can also be compared with surrounding resource sets to verify if the currently found resource set is indeed the largest.

Referring to FIG 3, the necessary resource sets before sorting are, in order as illustrated, necessary resource sets A, B, and C. After sorting, the necessary resource sets are, in order as illustrated, necessary resource sets C, B, and A.

In an embodiment of the application, when the allocation server 120 acquires the necessary resource set with the largest available resource remainder from among the multiple necessary resource sets as the first set, it can perform the following operation:

- select the first from the sorted multiple necessary resource sets as the first set. This allows for a quick selection of the necessary resource set with the largest available resource remainder, saving search time.

As shown in FIG 3, from the sorted necessary resource sets C, B, and A, C is directly selected as the first set.

In an embodiment of the application, after deducting the available resource remainder of the first set to zero, the allocation server 120 also performs the following operation:

- multiple necessary resource sets are sorted in descending order based on their updated available resource remainders.

As shown in FIG 4, after the first allocation shown in FIG 3, an available resource remainder of necessary resource set C changes from 30 to 0. After re-sorting, the multiple necessary resource sets are, in order, necessary resource sets B, A, and C.

In the next allocation, the necessary resource set B is directly taken as the first set.

In an embodiment of the application, the allocation server 120 is also configured to be responsible for handling resource allocation failures. Specifically, when the remaining demand for necessary resources required by the target process exceeds the total available resources across multiple necessary resource sets, the allocation server 120 is configured to generate the allocation failure information. This result is then returned to the background allocation component 110, alerting it to the allocation failure. This mechanism ensures that the system can promptly handle situations of insufficient resources, preventing tasks from waiting indefinitely when no resources are available for allocation.

During the resource allocation process, to quickly identify the necessary resource set with the largest available resource remainder, the allocation server 120 is configured to employ various sorting methods to organize the multiple necessary resource sets. These methods include, but are not limited to, bubble sort, insertion sort, selection sort, shell sort, merge sort, quick sort, heap sort, and counting sort. Depending on the specific application scenario and performance requirements, the most suitable sorting method can be chosen to enhance the efficiency of resource allocation.

The resource allocation approach outlined in this invention applies to multiple types of necessary resources, specifically comprising line resources, server resources, and telephone soft-switch platform resources (such as Freeswitch). These resources play a crucial role in intelligent voice services, and through effective resource allocation, their utilization can be optimized, thereby improving business processing efficiency.

After resource allocation is complete, to facilitate subsequent adjustments to the resource requirements of the target process, the allocation server 120 is configured to sort the multiple necessary resource sets based on the actual amount of resources called by the target process within each set. Uncalled resource sets are excluded from this sorting process. This strategy allows for the preferential release of resources from the least utilized sets when partial resource releases are necessary, maintaining resource concentration and reducing the probability of resource switching.

As illustrated in FIG 5, assuming the necessary resource sets are sorted in ascending order based on the actual resource usage by the target process after allocation, with Set B, Set C, and Set A being the order. When 10 units of resources need to be released, the system prioritizes releasing resources from Set B due to its minimal resource usage. If the resources released from Set B are insufficient (e.g., when 20 units of resources need to be released), the system continues to release resources from the next least utilized set (such as Set C) until the release requirement is met. This strategic approach to resource release ensures efficiency and orderliness in resource management.

The system in the embodiment of the application can quickly allocate resources, with tasks concentrated in relatively few resources. When resource failures occur, the number of affected tasks is relatively small. Each allocation prioritizes the resource with the largest available resource remainder, which also serves to balance the load.

As shown in FIG 6, an embodiment of the application also provides a method for allocating necessary resources for the audio data processing of conversations, comprising the following steps:

At step S210, obtaining the remaining demand for necessary resources required by the target process and multiple necessary resource sets associated with the target process.

At step S230, generating the total available resources of the multiple necessary resource sets based on the available resource remainder of each set.

At step S240, when the remaining demand for necessary resources required by the target process is less than or equal to the total available resources of the multiple necessary resource sets, repeat steps S241 to S243 until the remaining demand for necessary resources required by the target process is zero:

At step S241, selecting the necessary resource set with the largest available resource remainder from the multiple necessary resource sets as the first set.

At step S242, if the available resource remainder of the first set is greater than or equal to the remaining demand for necessary resources required by the target process, allocating the target process to use said first set, and deducting the available resource remainder of said first set based on the remaining demand for necessary resources required by the target process demand, and deducting the remaining demand for necessary resources required by said target process to zero.

At step S243, if the available resource remainder of the first set is less than the remaining demand for necessary resources required by the target process, allocating the entire available resource remainder of the first set to the target process, and deducting the remaining demand for necessary resources required by said target process based on the available resource remainder, and deducting the available resource remainder of said first set to zero.

As shown in FIG 2A, multiple necessary resource sets include necessary resource set A, necessary resource set B, and necessary resource set C. The available resource remainder of necessary resource set A is 10, the available resource remainder of necessary resource set B is 20, and the available resource remainder of necessary resource set C is 30. The total available resources are $30+20+10=60$. When the remaining demand for necessary resources for the target process is 70, successful allocation cannot be achieved because the remaining demand for necessary resources exceed the total available resources. As illustrated in FIG 2B, when the remaining demand for necessary resources for the target process are 40, allocation can proceed because the remaining demand for necessary resources is less than the available resource remainder. Among the necessary resource sets A, B, and C,

necessary resource set C with the largest available resource remainder is prioritized for allocation. However, since the available resource remainder of 30 in necessary resource set C is less than the remaining demand for necessary resources, the allocation cannot be completed in one go. All resources from necessary resource set C are allocated to the target process, and the available resource remainder of necessary resource set C is updated to zero after deduction. The current remaining demand for necessary resources for the target process become $40-30=10$. Among the remaining necessary resource sets, necessary resource set B with the largest available resource remainder is selected for allocation. The available resource remainder of necessary resource set B is 20 units, which is greater than the remaining demand for necessary resources of 10 units. Therefore, 10 units of resources from necessary resource set B are allocated to the target process, and the available resource remainder of necessary resource set B is updated to $20-10=10$. The current remaining demand for necessary resources for the target process drop to zero, indicating the end of allocation.

The allocation method in the embodiment of the application determines whether the total available resources from multiple necessary resource sets can meet the remaining demand for necessary resources for the target process. If so, it selects the necessary resource set with the largest available resource remainder for allocation. If the available resource remainder of a single necessary resource set meets the required necessary resources for the target process, the task is arranged to fully utilize the resources in that necessary resource set. If the necessary resource set with the largest available resource remainder cannot meet the target process's requirements, all resources from that necessary resource set are allocated to the target process. The process is repeated, with allocation continuing from the remaining necessary resource set with the largest available resource remainder until allocation is complete. This approach minimizes the number of allocation attempts, resulting in high efficiency. Additionally, task allocation is centralized, with fewer conversation audio data processing tasks corresponding to each necessary resource set. In case of resource failures, the impact on conversation audio data processing tasks is minimized, and switching efficiency is high.

In an embodiment of the application, as shown in FIG 7, before generating the total amount of available resources from multiple necessary resource sets based on the available resource remainder of each necessary resource set, the method comprises:

- 5 At step S220, sorting the multiple necessary resource sets according to the descending order of their available resource remainder.

Sorting the necessary resource sets facilitates finding the resource set with the largest available resource remainder. After finding the resource set with the largest available
10 resource remainder, it can also be compared with surrounding resource sets to verify whether the currently found resource set is indeed the largest.

As shown in FIG 3, the necessary resource sets before sorting are, in order, necessary resource sets A, B, and C. The necessary resource sets after sorting are, in order,
15 necessary resource sets C, B, and A.

In an embodiment of the application, obtaining the necessary resource set with the largest available resource remainder from among the multiple necessary resource sets as the first set comprises:

20

- selecting the first from the sorted multiple necessary resource sets as the first set.

As shown in FIG 3, from the sorted necessary resource sets C, B, and A, C is directly selected as the first set.

25

In an embodiment of the application, after deducting the available resource remainder of the first set to zero, the method further comprises:

- sorting the multiple necessary resource sets according to the descending order of
30 their updated available resource remainder.

As shown in FIG 4, after the first allocation, the available resource remainder of necessary resource set C changes from 30 to 0. After re-sorting, the multiple

necessary resource sets are, in order, necessary resource sets B, A, and C. In the next allocation, necessary resource set B will be directly selected as the first set.

In an embodiment of the application, as shown in FIG 8, the method further comprises:

5

At step S250, when the remaining demand for necessary resources required by the target process exceeds the total available resources of the multiple necessary resource sets, generating the allocation failure information.

10 When it is determined that the remaining demand for necessary resources required by the target process exceeds the total available resources of said multiple necessary resource sets, it indicates that full allocation is not possible, and the available resource remainder cannot meet the usage demand. In some embodiment, to ensure that the target process can be partially used, all available resource remainder may also be
15 allocated to the target process, which can be invoked using separate threads.

In an embodiment of the application, the sorting method comprises any one of the following methods: bubble sort, insertion sort, selection sort, Shell sort, merge sort, quick sort, heap sort, and counting sort.

20

In an embodiment of the application, the types of necessary resources include: line resources, server resources, and telephone softswitch platform resources.

In an embodiment of the application, as shown in FIG 9, after determining that the
25 remaining demand for necessary resources required by the target process is zero, the method further comprises:

At step S260, sorting the multiple necessary resource sets based on the ascending order of the actual amount of resources called by the target process in each necessary
30 resource set, wherein uncalled necessary resource sets are not sorted.

As shown in FIG 5, after allocation is complete, the necessary resource sets are sorted in ascending order based on the actual amount of resources called by the target process. The sorted necessary resource sets are, in order, necessary resource sets

B, C, and A. When partial resource release is required, such as releasing 10 resource units, the resources occupied in necessary resource set B are released first. After release, necessary resource set B is no longer associated with the target process. Even if necessary resource set B fails, it will not affect the execution of the target process. If 10 occupied resource units are released from C first, the occupied resource sets still include necessary resource sets C and B. Failure of either necessary resource set B or C will affect the execution of the target process, thereby increasing the probability of switching resources during the target process. When 20 resource units need to be released, priority is given to releasing from necessary resource set B. Since releasing 10 resource units is not sufficient, the smallest occupied resource set is selected from the remaining occupied resource sets for release, that is, another 10 resource units are released from necessary resource set C to meet the requirement.

An embodiment of the application can be applied to intelligent voice services. As shown in FIG 10, the specific implementation plan is as follows:

At step S310, obtaining the remaining available concurrency of all freeswitch and servers when allocating the necessary resources for intelligent voice dialing tasks.

At step S320, sorting the same resources in descending order based on the remaining available concurrency.

At step S330, calculating whether the remaining available concurrency meets the demand of the dialing task. If not, start allocating the next task.

At step S340, comparing the resources required for the task with the resource with the highest remaining available concurrency. If the available concurrency of the resource with the highest remaining available concurrency is greater than the resources required for the task, allocate the task to that resource, deduct its available quantity, and then re-sort. If the available quantity of the resource with the highest remaining available concurrency is less than the resources required for the task, allocate all the remaining available resources of that type to the task and find the next resource with the highest available concurrency, repeating the above operation until allocation is complete.

The method in an embodiment of the application allows for rapid resource allocation, with tasks concentrated on specific resources. When resource failures occur, the number of affected tasks is relatively small. Each allocation starts with the resource
5 with the highest available resource remainder, which also provides a certain load balancing function.

In an embodiment of the application, as shown in FIG 11, a necessary resource allocation device 1100 for dialogue audio data processing is presented, comprising:
10 an allocation preparation module 1110, a total available resource management module 1130, and a necessary resource allocation execution module 1140.

The allocation preparation module 1110 is configured to obtain the remaining demand for necessary resources required by the target process and multiple necessary
15 resource sets associated with the target process.

The total available resource management module 1120 is configured to generate the total available resources of the multiple necessary resource sets based on the available resource remainder of each necessary resource set.
20

The necessary resource allocation execution module 1140 is configured to, when the remaining demand for necessary resources required by the target process is less than or equal to the total available resources of the multiple necessary resource sets, cyclically perform the following steps until the remaining demand for necessary
25 resources required by the target process is zero:

- selecting the necessary resource set with the largest available resource remainder from said multiple necessary resource sets as the first set;

- 30 - if the available resource remainder of the first set is greater than or equal to the remaining demand for necessary resources required by the target process, allocating the target process to use said first set, and deducting the available resource remainder of said first set based on the remaining demand for necessary resources required by

the target process demand, and deducting the remaining demand for necessary resources required by said target process to zero;

- if the available resource remainder of the first set is less than the remaining demand for necessary resources required by the target process, allocating the entire available resource remainder of the first set to the target process, and deducting the remaining demand for necessary resources required by said target process based on the available resource remainder, and deducting the available resource remainder of said first set to zero.

10

In an embodiment of the application, a sorting module 1120 is also included, which is configured to sort the multiple necessary resource sets according to the descending order of available resource remainder of each necessary resource set.

15 In an embodiment of the application, the necessary resource allocation execution module 1140 is specifically configured to select the first from the sorted multiple necessary resource sets as the first set.

20 In an embodiment of the application, the necessary resource allocation execution module 1140 is further configured to sort the multiple necessary resource sets based on the descending order of available resource remainder of each updated necessary resource set.

25 In an embodiment of the application, a failure result information generation module 1150 is also included, configured to generate a failure allocation result when determining that the remaining demand for necessary resources required for the target process exceeds the total available resources of the multiple necessary resource sets.

30 In an embodiment of the application, the sorting method comprises any one of the following methods: bubble sort, insertion sort, selection sort, shell sort, merge sort, quick sort, heap sort, or counting sort.

In an embodiment of the application, the types of necessary resources include: line resources, server resources, and telephony softswitch platform resources.

In an embodiment of the application, an inverse sorting module 1160 is also included, configured to sort the multiple necessary resource sets based on the ascending order of the actual called resource amounts of the target process in each necessary resource set, after determining that the remaining demand for necessary resources required for the target process is zero, wherein the uncalled necessary resource sets are not sorted.

The device in the embodiment of the application can quickly allocate resources, and the tasks are relatively concentrated in resources. When resource failures occur, the affected tasks are relatively small. Each allocation starts with the resource with the largest available resource remainder for priority allocation, which also has a certain load balancing function.

In an embodiment, the device for allocating necessary resources for a dialog audio data processing procedure can combine the characteristics of the method and system for allocating necessary resources for a dialog audio data processing procedure in any embodiment, and vice versa. Details are not repeated here.

In an embodiment of the application, an electronic device is provided, comprising: a processor and a memory storing computer programs. The processor is configured to execute the method for allocating necessary resources for a dialog audio data processing procedure in any embodiment of the application when running the computer programs.

FIG 12 illustrates a schematic diagram of an electronic device 1200 that can implement the methods or embodiment of the application. In some embodiment, it may include more or fewer electronic devices than shown. In some embodiment, it can be implemented using a single or multiple electronic devices. In some embodiment, it can be implemented using cloud-based or distributed electronic devices.

As shown in FIG 12, the electronic device 1200 comprises a central processing unit (CPU) 1201, which can perform various appropriate operations and processing based on programs and/or data stored in read-only memory (ROM) 1202 and/or programs

and/or data loaded from the storage section 1208 into random access memory (RAM) 1203. The CPU 1201 can be a multi-core processor or include multiple processors. In some embodiment, the CPU 1201 can include a general-purpose main processor and one or more special coprocessors, such as a graphics processing unit (GPU), neural network processor (NPU), digital signal processor (DSP), and so on. Various programs and data required for the operation of the electronic device 1200 are also stored in the RAM 1203. The CPU 1201, ROM 1202, and RAM 1203 are connected to each other via a bus 1204. An input/output (I/O) interface 1205 is also connected to the bus 1204.

10 The above-mentioned processor and memory are jointly used to execute the program stored in the memory. When the program is executed by a computer, it can achieve the steps or functions of the method for allocating necessary resources for a dialog audio data processing procedure described in the above embodiment.

15 The following components are connected to the I/O interface 1205: an input section 1206 comprising a keyboard, mouse, touchscreen, etc.; an output section 1207 comprising a cathode ray tube (CRT), liquid crystal display (LCD), speakers, etc.; a storage section 1208 comprising a hard disk, etc.; and a communication section 1209 comprising a network interface card such as a LAN card, modem, etc. The communication section 1209 performs communication processing via a network such as the Internet. A drive 1210 is also connected to the I/O interface 1205 as needed. A removable medium 1211, such as a magnetic disk, optical disk, magneto-optical disk, semiconductor memory, etc., is installed on the drive 1210 as needed, so that computer programs read from it can be installed into the storage section 1208 as required. FIG 12 only schematically shows some components and does not mean that the computer system 1200 only comprises the components shown in FIG 12.

In some embodiment, the electronic device 1020 refers to a mobile terminal, comprising a mobile phone, vehicle terminal, smart TV, etc. Taking a mobile phone as an example, the electronic device 1200 also comprises a touch-sensitive display screen, external speakers, a gyroscope, camera, 4G/5G antenna, and other device modules.

The systems, devices, modules, or units clarified in the above embodiment can be implemented by a computer or its associated components. The computer can be, for example, a mobile terminal, smartphone, personal computer, laptop, in-vehicle human-computer interaction device, personal digital assistant, media player, navigation device, game console, tablet computer, wearable device, smart TV, Internet of Things system, smart home, industrial computer, server, or a combination of these.

Although not shown, in the embodiment of the application, a storage medium storing computer programs is provided. The computer programs are configured to execute the task scheduling and monitoring method of any embodiment of the application when run.

The storage medium of the embodiment of the application comprises permanent and non-permanent, removable and non-removable items that can be implemented by any method or technology to store information. Examples of storage medium include, but are not limited to, Phase-change Random Access Memory (PRAM), Static Random Access Memory (SRAM), Dynamic Random Access Memory (DRAM), other types of Random Access Memory (RAM), Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), Flash memory or other memory technologies, Compact Disk Read-Only Memory (CD-ROM), Digital Versatile Disk (DVD) or other optical storage, magnetic cassette tapes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other non-transmission medium that can be used to store information accessible by a computing device.

The methods, programs, systems, devices, etc., of the embodiment of the application can be executed or implemented in a single or multiple networked computers, and can also be practiced in a distributed computing environment. In the embodiment of this specification, in these distributed computing environments, tasks can be performed by remote processing devices connected via a communication network.

Those skilled in the art should understand that the embodiment of this specification can be provided as methods, systems, or computer program products. Therefore, those skilled in the art can conceive that the implementation of the functional modules/units or controllers and related method steps elucidated in the above

embodiment can be achieved through a combination of software, hardware, and software/hardware.

5 Unless explicitly stated, the actions or steps of the methods and programs recorded in the embodiment of the application do not necessarily need to be performed in a specific order and can still achieve the desired results. In some embodiment, multitasking and parallel processing are also possible or may be advantageous.

10 In the application, multiple embodiment of the application are described, but for brevity, the descriptions of each embodiment are not exhaustive, and similar or identical features or parts between various embodiment may be omitted. In this article, "an embodiment", "some embodiment", "examples", "specific examples", or "some examples" refer to at least one embodiment or example according to the application, rather than all embodiment. The above terms do not necessarily refer to the same
15 embodiment or example. Without contradicting each other, those skilled in the art can combine and integrate different embodiment or examples and the characteristics of different embodiment or examples described in this specification.

20 Throughout this specification and claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated integer or group of integers or steps but not the exclusion of any other integer or group of integers.

25 Persons skilled in the art will appreciate that numerous variations and modifications will become apparent. All such variations and modifications which become apparent to persons skilled in the art, should be considered to fall within the spirit and scope that the invention broadly appearing before described.

CLAIMS

1. A method for necessary resource allocation during dialog audio data processing,
5 characterized in that the method comprises the following steps:
obtaining the remaining demand for necessary resources required by the target
process and multiple necessary resource sets associated with said process;
generating the total available resources of said multiple necessary resource sets
based on the available resource remainder of each set;
10 when the remaining demand for necessary resources required by the target process
is less than or equal to the total available resources of said multiple necessary
resource sets, cyclically executing the following steps until the remaining demand for
necessary resources reaches zero:
selecting the necessary resource set with the largest available resource remainder
15 from said multiple necessary resource sets as the first set;
if the available resource remainder of the first set is greater than or equal to the
remaining demand for necessary resources required by the target process, allocating
the target process to use said first set, and deducting the available resource remainder
of said first set based on the remaining demand for necessary resources required by
20 the target process demand, and deducting the remaining demand for necessary
resources required by said target process to zero;
if the available resource remainder of the first set is less than the remaining demand
for necessary resources required by the target process, allocating the entire available
resource remainder of the first set to the target process, and deducting the remaining
25 demand for necessary resources required by said target process based on the
available resource remainder, and deducting the available resource remainder of said
first set to zero.
2. The method according to claim 1, characterized in that before generating the total
30 available resources of said multiple necessary resource sets, sorting said multiple
necessary resource sets in descending order based on their available resource
remainders.

3. The method according to claim 2, characterized in that selecting the necessary resource set with the largest available resource remainder as the first set comprises choosing the first set from the sorted multiple necessary resource sets.

5 4. The method according to claim 3, characterized in that after deducting the available resource remainder of said first set to zero, sorting said multiple necessary resource sets again in descending order based on their updated available resource remainders.

10 5. The method according to claim 1, characterized in that when the remaining demand for necessary resources required by the target process is greater than the total available resources of said multiple necessary resource sets, generating the allocation failure information.

15 6. The method according to claim 1, characterized in that the types of necessary resources include line resources, server resources, and freeswitch resources.

20 7. The method according to claim 1, characterized in that after the remaining demand for necessary resources required by the target process reaches zero, sorting said multiple necessary resource sets in ascending order based on the actual amount of resources called by the target process, with unset resources not being sorted.

25 8. A device for necessary resource allocation during dialog audio data processing, characterized in that the device comprises an allocation preparation module, a total available resource management module, and a necessary resource allocation execution module, wherein:

the allocation preparation module is configured to obtain the remaining demand for necessary resources required by the target process and multiple necessary resource sets associated with said process;

30 the total available resource management module is configured to generate the total available resources of said multiple necessary resource sets based on the available resource remainder of each set;

the necessary resource allocation execution module is configured to, when the remaining demand for necessary resources required by the target process is less than or equal to the total available resources of said multiple necessary resource sets,

cyclically execute the following steps until the remaining demand for necessary resources reaches zero:

selecting the necessary resource set with the largest available resource remainder from said multiple necessary resource sets as the first set;

- 5 if the available resource remainder of the first set is greater than or equal to the remaining demand for necessary resources required by the target process, allocating the target process to use said first set, and deducting the available resource remainder of said first set based on the remaining demand for necessary resources required by the target process demand, and deducting the remaining demand for necessary
- 10 resources required by said target process to zero;
- if the available resource remainder of the first set is less than the remaining demand for necessary resources required by the target process, allocating the entire available resource remainder of the first set to the target process, and deducting the remaining demand for necessary resources required by said target process based on the
- 15 available resource remainder, and deducting the available resource remainder of said first set to zero.

9. A computer-readable storage medium, characterized in that stored a computer program, wherein said computer program executed by a processor and implements
- 20 any of the methods according to claims 1-7.

10. An electronic device, characterized in that comprises a processor and a memory storing a computer program, wherein said processor is configured to execute any of the methods according to claims 1-7 when running said computer program.

25