# IMAGE SCANNING METHOD AND APPARATUS, AND ELECTRONIC DEVICE

**[0001]** This disclosure claims priority to Chinese Patent Application No. 202410290251.1, filed with the China National Intellectual Property Administration on March 13, 2024, and entitled "IMAGE SCANNING METHOD AND APPARATUS, AND ELECTRONIC DEVICE", the entire contents of which are incorporated herein by reference.

## TECHNICAL FIELD

**[0002]** The present disclosure relates to the field of cloud security, and more particularly, to an image scanning method and apparatus, and an electronic device.

## BACKGROUND

**[0003]** Container image security solutions, especially static scanning schemes for image artifacts, are already relatively mature. The static scanning scheme identifies risks by scanning various files included in image files. With the enrichment of scanning capabilities, scanning rules are also increasing, and the requirements for scanning computing capabilities and server capabilities are also becoming higher and higher.

**[0004]** However, the coupling of execution scripts and rules in the scanning scheme, and the coupling of the business module and the computing module, result in excessively long scanning duration and poor performance.

## SUMMARY

**[0005]** The present disclosure provides an image scanning method and apparatus, and an electronic device, providing an independent computing unit with elastic computing power based on a serverless architecture, which improves scanning performance and reduces scanning duration.

**[0006]** In a first aspect, the present disclosure provides an image scanning method, including:

obtaining a file to be detected in an image file;

splitting the file to be detected into a plurality of batches, and generating a scanning sub-task corresponding to each batch;

for the scanning sub-task of each batch, triggering a computing unit to load a scanning rule corresponding to the scanning sub-task, and performing, according to the scanning rule, computation corresponding to the scanning sub-task, where the computing unit is implemented through a Function Compute FC service;

merging computation results corresponding to the scanning sub-tasks to obtain an image scanning result.

**[0007]** In an implementation, the obtaining the file to be detected in the image file includes:

obtaining the image file from an image repository;

obtaining a pre-filtering rule, and filtering the image file based on the pre-filtering rule to determine the file to be detected.

**[0008]** In an implementation, after the obtaining the file to be detected in the image file, the method further includes:

transmitting the file to be detected to an intermediate storage space other than a service server.

**[0009]** In an implementation, the splitting the file to be detected into the plurality of batches, and generating the scanning sub-task corresponding to each batch includes:

splitting the file to be detected into the plurality of batches according to a pre-configured concurrency strategy, and generating the scanning sub-task corresponding to each batch.

**[0010]** In an implementation, the for the scanning sub-task of each batch, triggering the computing unit to load the scanning rule corresponding to the scanning sub-task, and performing, according to the scanning rule, the computation corresponding to the scanning sub-task includes:

for the scanning sub-task of each batch, invoking an input component of the computing unit to trigger a computing execution component of the computing unit, loading, by the computing execution component, the scanning rule corresponding to the scanning sub-task, performing, according to the scanning rule, the computation corresponding to the scanning sub-task, and outputting, by an output component of the computing unit, the computation result corresponding to the scanning sub-task.

**[0011]** In an implementation, the merging the computation results corresponding to the scanning sub-tasks to obtain the image scanning result includes:

distributing the computation results corresponding to respective scanning sub-tasks to recheck engines corresponding to the scanning sub-tasks, and adding an asynchronous-to-synchronous flag to some or all of the scanning sub-tasks;

after obtaining recheck results corresponding to all scanning sub-tasks to which the asynchronous-to-synchronous flag is added, merging respective recheck results to obtain the image scanning result.

**[0012]** In an implementation, the recheck result includes a detection item name, a detection result, and a detection duration corresponding to the scanning sub-task.

**[0013]** In a second aspect, the present disclosure provides an image scanning apparatus, including:

an obtaining module, configured to obtain a file to be detected in an image file;

a splitting module, configured to split the file to be detected into a plurality of batches, and generate a scanning sub-task corresponding to each batch;

a computing module, configured to, for the scanning sub-task of each batch, trigger a computing unit to load a scanning rule corresponding to the scanning sub-task, and perform, according to the scanning rule, computation corresponding to the scanning sub-task, where the computing unit is implemented through a Function Compute FC service;

a synchronization module, configured to merge computation results corresponding to the scanning sub-tasks to obtain an image scanning result.

**[0014]** In an implementation, the obtaining module is configured to:

obtain the image file from an image repository;

obtain a pre-filtering rule, and filter the image file based on the pre-filtering rule to determine the file to be detected.

**[0015]** In an implementation, the obtaining module is configured to:

transmit the file to be detected to an intermediate storage space other than a service server.

**[0016]** In an implementation, the splitting module is configured to:

split the file to be detected into the plurality of batches according to a pre-configured concurrency strategy, and generate the scanning sub-task corresponding to each batch.

**[0017]** In an implementation, the computing module is configured to:

for the scanning sub-task of each batch, invoke an input component of the computing unit to trigger a computing execution component of the computing unit, load, by the computing execution component, the scanning rule corresponding to the scanning sub-task, perform, according to the scanning rule, the computation corresponding to the scanning sub-task, and output, by an output component of the computing unit, the computation result corresponding to the scanning sub-task.

**[0018]** In an implementation, the synchronization module is configured to:

distribute the computation results corresponding to respective scanning sub-tasks to recheck engines corresponding to the scanning sub-tasks, and add an asynchronous-to-synchronous flag to some or all of the scanning sub-tasks;

after obtaining recheck results corresponding to all scanning sub-tasks to which the asynchronous-to-synchronous flag is added, merge respective recheck results to obtain the image scanning result.

**[0019]** In an implementation, the recheck result includes a detection item name, a detection result, and a detection duration corresponding to the scanning sub-task.

**[0020]** In a third aspect, the present disclosure provides an electronic device, including: a memory and a processor;

the memory is configured to store a computer program;

the processor is configured to execute the computer program stored in the memory, where when the computer program is executed, the processor is caused to perform the method according to the first aspect.

**[0021]** In a fourth aspect, the present disclosure provides a computer-readable storage medium, a computer program is stored in the computer-readable storage medium, and when the computer program is executed by a processor, the processor is caused to perform the method according to the first aspect.

**[0022]** In a fifth aspect, the present disclosure provides a computer program product, including a computer program, and when the computer program is executed by a processor, the method according to the first aspect is implemented.

**[0023]** In the image scanning method and apparatus, and the electronic device provided by the present disclosure, by splitting the file to be detected in the image file into a plurality of batches and generating a scanning sub-task corresponding to each batch; for the scanning sub-task of each batch, triggering a computing unit implemented based on an FC service to load a scanning rule corresponding to the scanning sub-task, and performing, according to the scanning rule, computation corresponding to the scanning sub-task, complete separation of business code and computing logic is achieved, computing scripts and scanning rules are separated, and by using an independent computing unit with computing capability and elastic scaling capability based on the FC service, the most performance-consuming computing steps in scanning are split into a plurality of concurrent tasks, thereby comprehensively improving the performance of the scanning process.

## BRIEF DESCRIPTION OF DRAWINGS

**[0024]** In order to more clearly illustrate the technical solutions in the embodiments of the present disclosure or the prior art, the accompanying drawings used in the description of the embodiments or the prior art will be briefly introduced below. Obviously, the accompanying drawings in the following description are some embodiments of the present disclosure. For those skilled in the art, other drawings can also be obtained based on these drawings without creative work.

**[0025]** FIG. 1 is a schematic diagram of an architecture of an image scanning solution according to an embodiment of the present disclosure.

**[0026]** FIG. 2 is a schematic flowchart of an image scanning method according to an embodiment of the present disclosure.

**[0027]** FIG. 3 is a schematic structural diagram of an image scanning apparatus according to an embodiment of the present disclosure.

**[0028]** FIG. 4 is a schematic block diagram of an electronic device according to an embodiment of the present disclosure.

## DETAILED DESCRIPTION OF EMBODIMENTS

**[0029]** To make the objectives, technical solutions, and advantages of the embodiments of the present disclosure clearer, the technical solutions in the embodiments of the present disclosure will be clearly and completely described below with reference to the accompanying drawings in the embodiments of the present disclosure. Obviously, the described embodiments are some, but not all, of the embodiments of the present disclosure. Based on the embodiments of the present disclosure, all other embodiments obtained by those of ordinary skill in the art without creative effort shall fall within the scope of protection of the present disclosure.

**[0030]** The terminology involved in the embodiments of the present disclosure is introduced below.

**[0031]** Computing unit: a stateless module that includes only three parts, namely input, output, and computing, is independent of the business system, does not participate in business logic, and is only responsible for performing complex computations (regular expression matching, format verification).

**[0032]** Serverless: a serverless architecture, a cloud computing architecture paradigm that enables developers to build and run applications on a cloud platform without managing the underlying server infrastructure. The cloud platform automatically handles the deployment, scaling, and management of applications, and automatically allocates and releases resources according to the needs of the applications. This architecture paradigm can provide higher development efficiency and lower operational costs, while possessing high scalability and elasticity.

**[0033]** Function Compute (Function Computing, FC): a cloud computing service model that enables developers to write and deploy application code in a serverless manner. In Function Compute, developers only need to write function code and upload it to the cloud platform, without needing to concern themselves with server management and maintenance. When a request triggers a function, the cloud platform automatically allocates resources to execute the function and returns the execution result.

**[0034]** Asynchronous scanning: many detection items in image scanning require a downstream engine to perform rechecking, and therefore the detection items all gradually flow back to the cloud after scanning is completed.

**[0035]** A traditional image scanning solution typically includes the following parts: rule management, rule execution, and data summarization and reporting. The rule execution part

involves pulling an image, decompressing the image, traversing files, and rule matching, where the rule matching includes a large amount of regular expression operations. To ensure scanning performance, multi-threaded stream processing is generally adopted. This solution has the following problems:

**[0036]** 1. High coupling and difficult to port: execution scripts are coupled with rules, and business code is coupled with computation logic; image data needs to be stored locally on a service server, making it difficult to be compatible with various complex cloud environments in a hybrid cloud, and affecting scanning performance.

**[0037]** 2. Scanning performance bottleneck: limited by the performance of a single service server and the upper limit of thread concurrency, the scanning duration of traditional image scanning is difficult to reduce, and as there are more and more rules, the scanning duration also increases accordingly.

**[0038]** 3. High stability risk: because the local service server needs to perform a large amount of regular expression computation, the pressure on a Central Processing Unit (CPU) is very high, and when reaching a scanning Queries-per-second (QPS) peak value, multi-threaded concurrent regular expression computation causes a CPU utilization rate of 100%.

**[0039]** 4. High idle cost: to cope with the scanning QPS peak value, a high-specification CPU, memory, and a large-capacity hard disk are required, and these configurations have extremely low utilization during scanning QPS off-peak periods, causing considerable resource idle costs.

**[0040]** 5. Unstable timeliness of results: scanning results for most detection items need to be rechecked by a recheck engine, and currently asynchronous detection result reporting is mostly adopted, that is, after rechecking of each detection item is completed, recheck results are reported asynchronously, resulting in unstable timeliness of results.

**[0041]** In view of this, an embodiment of the present disclosure proposes an image scanning solution. With reference to the architecture diagram shown in FIG. 1, the architectural core of this solution lies in completely separating business code and computation logic, establishing an independent computing unit having computing capability and elastic scaling capability, splitting the most performance-consuming computation steps in scanning into multiple concurrent tasks, and at the same time adapting, by a cloud scanner, the input and output and task scheduling of the computing unit. The details are as follows:

**[0042]** I. Computing unit of Serverless architecture has the following characteristics:

**[0043]** The computing unit is implemented through an FC service, uses a base image to initialize an FC computing instance, and has the elastic capability of creation and destruction within seconds, where the base image is a model obtained by abstracting various risk types that need to be detected and scanned.

**[0044]** The computing unit includes three logical units: a uniform-format input component, a computing execution component, and a uniform-format output component, where each computing unit is unaware of business data and does not depend on any business environment.

**[0045]** Rules and computation scripts are separated; the rules are operated independently, and the computation scripts are fully reused.

**[0046]** II. Cloud scanner has the following characteristics:

**[0047]** The cloud scanner supports independently operating scanning rules, which take effect in real time.

**[0048]** The cloud scanner supports flexible concurrency configuration, splitting a scanning task for a single image into multiple scanning sub-tasks according to the number of files in a single batch, the number of scanning rules, etc., where the scanning sub-tasks are executed in parallel and are finally aggregated into a complete image scanning result.

**[0049]** The cloud scanner implements asynchronous-to-synchronous conversion of results. After delivering computation results of the scanning sub-tasks to a downstream engine, the cloud scanner performs unified merging of recheck results reported by the downstream engine, and the cloud scanner performs task state transitions, implementing synchronous simulation of an asynchronous process.

**[0050]** FIG. 2 is a schematic flowchart of an image scanning method provided by an embodiment of the present disclosure. The execution entity of this method is a cloud scanner. As shown in FIG. 2, the method includes the following.

**[0051]** Step S201: obtaining a file to be detected in an image file.

**[0052]** Optionally, with reference to the architecture of FIG. 1, the cloud scanner obtains the image file from an image repository; obtains a pre-filtering rule, and filters the image file based on the pre-filtering rule to determine the file to be detected. For example, the pre-filtering rule includes filtering on file size, path, and character strings. After decompressing the image file, the cloud scanner removes useless files from the decompressed image file based on the pre-filtering rule, thereby reducing the computation amount of image scanning.

**[0053]** Optionally, after obtaining the file to be detected in the image file, the method of the embodiment of the present disclosure further includes: transmitting the file to be detected to an intermediate storage space other than a service server.

**[0054]** For example, the file to be detected is stored in an object storage service (OSS) space, thereby avoiding occupation of the service server.

**[0055]** Step S202: splitting the file to be detected into a plurality of batches, and generating a scanning sub-task corresponding to each batch.

**[0056]** Optionally, the file to be detected is split into the plurality of batches according to a pre-

configured concurrency strategy, and the scanning sub-task corresponding to each batch is generated. For example, different scanning sub-tasks can correspond to the same or different detection items, such as vulnerability detection, limit detection, binary file detection, etc. The concurrency strategy may be determined by a user according to the number of files to be detected, scanning rules, etc., and can support flexible configuration. For example, the concurrency strategy can indicate the number of scanning sub-tasks that are executed in parallel. For example, there are 10,000 files to be detected in total, and the files to be detected are split into 100 batches, with 100 files in each batch.

**[0057]** Step S203: for the scanning sub-task of each batch, triggering a computing unit to load a scanning rule corresponding to the scanning sub-task, and performing, according to the scanning rule, computation corresponding to the scanning sub-task, where the computing unit is implemented through an FC service.

**[0058]** Each scanning sub-task is executed by one computing unit instance, and multiple computing unit instances can execute different scanning sub-tasks in parallel.

**[0059]** For the scanning sub-task of each batch, an input component of the computing unit is invoked to trigger a computing execution component of the computing unit, the scanning rule corresponding to the scanning sub-task is loaded by the computing execution component, computation corresponding to the scanning sub-task is performed according to the scanning rule, and then a computation result corresponding to the scanning sub-task is output by an output component of the computing unit.

**[0060]** The input component of the computing unit needs to obtain input parameters and input files. The input parameters can include a file index, a scanning rule index, etc., and the input files are obtained from the intermediate storage space based on the input parameters. The computing execution component can download, from independently operated scanning rules, the scanning rule corresponding to the scanning sub-task based on the scanning rule index.

**[0061]** Step S204: merging computation results corresponding to the scanning sub-tasks to obtain an image scanning result.

**[0062]** By performing unified merging of computation results of multiple scanning sub-tasks, asynchronous results are simulated as synchronous.

**[0063]** In a scenario where a recheck engine performs rechecking, computation results corresponding to respective scanning sub-tasks are distributed to the recheck engines corresponding to the scanning sub-tasks, and an asynchronous-to-synchronous flag is added to some or all of the scanning sub-tasks; after recheck results corresponding to all scanning sub-tasks to which the asynchronous-to-synchronous flag is added are obtained, respective recheck results are merged to obtain the image scanning result, achieving synchronous state simulation that unifies

the time points of the scanning task state and the data return state.

**[0064]** Optionally, the recheck result returned by the recheck engine includes a detection item name, a detection result, and a detection duration corresponding to the scanning sub-task. The cloud scanner merges the above results returned by respective recheck engines to complete the synchronous state change of scanning.

**[0065]** The image scanning method provided by the embodiment of the present disclosure completely separates business code and computation logic, separates computation scripts and scanning rules, establishes an independent computing unit which is based on an FC service and has computing capability and elastic scaling capability, and splits the most performance-consuming computation steps in scanning into multiple concurrent tasks, comprehensively improving the performance of the image scanning process, reducing scanning duration, and facilitating multi-cloud environment porting.

**[0066]** Specifically, by using the independent computing unit to decouple computing capability and business logic, multiple problems such as high coupling, performance risk, and high cost are solved. The independent computing unit has the elastic capability of creation and destruction within seconds, and through multi-task concurrent splitting, maximum parallel processing is achieved to implement high-performance scanning. At the same time, the independent computing unit is only responsible for computation of the current task and does not consume the CPU of a cloud service server, so the CPU and storage space within the computing unit can be fully used without generating other stability risks. Because the cloud service server no longer needs to perform complex computation, the CPU and storage configuration can be appropriately reduced, saving costs. Under a pay-as-you-go usage model, resources are fully utilized during peak periods, and no additional costs are incurred during idle periods.

**[0067]** Furthermore, through decoupling computation and business, scanning rules are processed in the business flow, and actual computation is performed in the independent computing unit. When the scanning rules need to be changed or upgraded, only changes to the rule part need to be focused on, and modifications take effect in real time, without requiring a cumbersome development and release process, achieving efficient iteration and real-time upgrading.

**[0068]** Moreover, the cloud scanner and a downstream recheck engine implement synchronous state simulation of asynchronous processing results, unifying the time points of the scanning task state and the data return state, achieving unification of the timeliness of state and result.

**[0069]** FIG. 3 is a schematic structural diagram of an image scanning apparatus provided by an embodiment of the present disclosure. As shown in FIG. 3, the image scanning apparatus 300 includes:

an obtaining module 301, configured to obtain a file to be detected in an image file;

a splitting module 302, configured to split the file to be detected into a plurality of batches, and generate a scanning sub-task corresponding to each batch;

a computing module 303, configured to, for the scanning sub-task of each batch, trigger a computing unit to load a scanning rule corresponding to the scanning sub-task, and perform, according to the scanning rule, computation corresponding to the scanning sub-task, where the computing unit is implemented through a Function Compute FC service;

a synchronization module 304, configured to merge computation results corresponding to the scanning sub-tasks to obtain an image scanning result.

**[0070]** In an implementation, the obtaining module 301 is configured to:

obtain the image file from an image repository;

obtain a pre-filtering rule, and filter the image file based on the pre-filtering rule to determine the file to be detected.

**[0071]** In an implementation, the obtaining module 301 is configured to:

transmit the file to be detected to an intermediate storage space other than a service server.

**[0072]** In an implementation, the splitting module 302 is configured to:

split the file to be detected into the plurality of batches according to a pre-configured concurrency strategy, and generate the scanning sub-task corresponding to each batch.

**[0073]** In an implementation, the computing module 303 is configured to:

for the scanning sub-task of each batch, invoke an input component of the computing unit to trigger a computing execution component of the computing unit, load, by the computing execution component, the scanning rule corresponding to the scanning sub-task, perform, according to the scanning rule, computation corresponding to the scanning sub-task, and output the computation result corresponding to the scanning sub-task by an output component of the computing unit.

**[0074]** In an implementation, the synchronization module 304 is configured to:

distribute computation results corresponding to respective scanning sub-tasks to recheck engines corresponding to the scanning sub-tasks, and add an asynchronous-to-synchronous flag to some or all of the scanning sub-tasks;

after obtaining recheck results corresponding to all scanning sub-tasks to which the asynchronous-to-synchronous flag is added, merge respective recheck results to obtain the image scanning result.

**[0075]** In an implementation, the recheck result includes a detection item name, a detection result, and a detection duration corresponding to the scanning sub-task.

**[0076]** The apparatus of the embodiment of the present disclosure is used to execute the image scanning method in the foregoing embodiments, and the implementation principles and technical

effects thereof are similar, which will not be repeated here.

**[0077]** FIG. 4 is a schematic block diagram of an electronic device provided by an embodiment of the present disclosure. As shown in FIG. 4, the electronic device 400 can include at least one processor 401, configured to implement the image scanning method provided by the embodiment of the present disclosure.

**[0078]** Optionally, the electronic device 400 further includes at least one memory 402, configured to store program instructions and/or data. The memory 402 is coupled to the processor 401. The coupling in the embodiments of the present disclosure is an indirect coupling or communication connection between apparatuses, units, or modules, which may be electrical, mechanical, or in other forms, and is used for information exchange between apparatuses, units, or modules. The processor 401 can operate in cooperation with the memory 402. The processor 401 can execute program instructions stored in the memory 402. At least one of the at least one memory is included in the processor.

**[0079]** Optionally, the electronic device 400 further includes a communication interface 403, configured to communicate with other devices through a transmission medium, so that the electronic device 400 can communicate with other devices. The communication interface 403 is, for example, a transceiver, an interface, a bus, a circuit, or an apparatus capable of implementing a transceiving function. The processor 401 can use the communication interface 403 to send and receive data and/or information, and is used to implement the method provided by the embodiment of the present disclosure. For details, reference is made to the detailed description in the foregoing embodiments, which will not be repeated here.

**[0080]** In the embodiment of the present disclosure, the specific connection medium among the processor 401, the memory 402, and the communication interface 403 is not limited. In the embodiment of the present disclosure, FIG. 4 shows the processor 401, the memory 402, and the communication interface 403 connected through a bus 404. The bus 404 is represented by a bold line in FIG. 4, and the connection manner between other components is merely a schematic illustration and is not intended to be limiting. The bus is divided into an address bus, a data bus, a control bus, etc. For ease of representation, only one bold line is used in FIG. 4, but this does not mean there is only one bus or only one type of bus.

**[0081]** It should be understood that the processor in the embodiment of the present disclosure is an integrated circuit chip having signal processing capability. In the implementation process, the steps of the foregoing method embodiments are completed by an integrated logic circuit of hardware in the processor or by instructions in the form of software. The processor is a general-purpose processor, a digital signal processor (DSP), an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA) or other programmable logic device, a discrete

gate or transistor logic device, or a discrete hardware component. The processor can implement or execute the methods, steps, and logic block diagrams disclosed in the embodiments of the present disclosure. The general-purpose processor is a microprocessor or the processor is also any conventional processor, etc. The steps of the method disclosed in the embodiments of the present disclosure are directly embodied as being executed and completed by a hardware decoding processor, or are executed and completed by a combination of hardware and software modules in the decoding processor. A software module is located in a mature storage medium in the art such as a random access memory, a flash memory, a read-only memory, a programmable read-only memory, an electrically erasable programmable memory, or a register. The storage medium is located in the memory, and the processor reads information in the memory and completes the steps of the foregoing method in combination with its hardware.

**[0082]** It should also be understood that the memory in the embodiment of the present disclosure is a volatile memory or a non-volatile memory, or can include both volatile and non-volatile memory. The non-volatile memory is a read-only memory (ROM), a programmable ROM (PROM), an erasable PROM (EPROM), an electrically erasable PROM (EEPROM), or a flash memory. The volatile memory is a random access memory (RAM), which is used as an external cache. By way of example and not limitation, many forms of RAM are available, such as a static RAM (SRAM), a dynamic RAM (DRAM), a synchronous DRAM (SDRAM), a double data rate SDRAM (DDR SDRAM), an enhanced SDRAM (ESDRAM), a synchlink DRAM (SLDRAM), and a Direct Rambus RAM (DR RAM). It should be noted that the memory of the systems and methods described herein is intended to include, but is not limited to, these and any other suitable types of memory.

**[0083]** The present disclosure also provides a computer-readable storage medium, where the computer-readable storage medium stores a computer program (which can also be referred to as code, or instructions). When the computer program is executed by a processor, a computer is caused to execute the method in any of the foregoing embodiments.

**[0084]** The present disclosure also provides a computer program product, including a computer program, where when the computer program is executed by a processor, the method in any of the foregoing embodiments is implemented.

**[0085]** The terms “unit” and “module” used in this specification are used to represent computer-related entities, hardware, firmware, a combination of hardware and software, software, or software in execution.

**[0086]** Those of ordinary skill in the art can appreciate that the various illustrative logical blocks and steps described in connection with the embodiments disclosed herein are implemented by electronic hardware, or a combination of computer software and electronic hardware. Whether

these functions are performed in hardware or software depends on the specific application and design constraints of the technical solution. Those skilled in the art can use different methods to implement the described functions for each specific application, but such implementation should not be considered to be beyond the scope of the present disclosure. In several embodiments provided in the present disclosure, it should be understood that the disclosed apparatus, device, and method are implemented in other manners. For example, the apparatus embodiments described above are merely schematic; for instance, the division of the units is merely a logical function division, and there are other division manners in actual implementation, for example, multiple units or components are combined or are integrated into another system, or some features are omitted or not executed. In addition, the mutual coupling, direct coupling, or communication connection shown or discussed is an indirect coupling or communication connection through some interfaces, apparatuses, or units, which may be electrical, mechanical, or in other forms.

**[0087]** A unit described as a separate component can or cannot be physically separated, and a component displayed as a unit can or cannot be a physical unit, that is, it can be located in one place, or distributed to multiple network units. Some or all of the units are selected according to actual needs to achieve the objective of the solution of this embodiment.

**[0088]** In addition, functional units in various embodiment of the present disclosure can be integrated into one processing unit, or each unit can physically exist separately, or two or more units can be integrated into one unit.

**[0089]** In the foregoing embodiments, the functions of each functional unit can be implemented entirely or partially by software, hardware, firmware, or any combination thereof. When implemented using software, the functions can be entirely or partially implemented in the form of a computer program product. The computer program product includes one or more computer instructions (programs). When the computer program instructions (programs) are loaded and executed on a computer, the process or functions according to the embodiments of the present disclosure are entirely or partially generated. The computer can be a general-purpose computer, a special-purpose computer, a computer network, or other programmable apparatus. The computer instructions can be stored in a computer-readable storage medium, or transmitted from one computer-readable storage medium to another computer-readable storage medium; for example, the computer instructions can be transmitted from a website site, computer, server, or data center to another website site, computer, server, or data center through a wired (for example, coaxial cable, optical fiber, digital subscriber line (DSL)) or wireless (for example, infrared, wireless, microwave, etc.) manner. The computer-readable storage medium can be any available medium that can be accessed by a computer or a data storage device, such as a server or a data center, that includes one or more available media integrated therein. The available medium can be a magnetic

medium (for example, a floppy disk, a hard disk, a magnetic tape), an optical medium (for example, a digital video disc (DVD)), or a semiconductor medium (for example, a solid state disk (SSD)), etc.

**[0090]** If the functions are implemented in the form of software functional units and sold or used as an independent product, they are stored in a computer-readable storage medium. Based on such understanding, the technical solution of the present disclosure essentially, or the part contributing to the prior art, or part of the technical solution can be embodied in the form of a software product. The computer software product is stored in a storage medium and includes several instructions for causing a computer device (which is a personal computer, a server, or a network device, etc.) to execute all or some of the steps of the methods in each embodiment of the present disclosure. The foregoing storage medium includes: a USB flash drive, a removable hard disk, a ROM, a RAM, a magnetic disk, an optical disk, and various other media that can store program code.

**[0091]** The user information (including but not limited to user device information, user personal information, etc.) and data (including but not limited to data used for analysis, stored data, displayed data, etc.) involved in the present disclosure are all information and data authorized by the user or fully authorized by all parties, and the collection, use, and processing of relevant data need to comply with the relevant laws, regulations, and standards of the relevant countries and regions, and corresponding operation entries are provided for the user to select authorization or refusal.

**[0092]** The foregoing descriptions are merely specific implementations of the present disclosure, but the scope of protection of the present disclosure is not limited thereto. Those skilled in the art familiar with the technical field can readily conceive of variations or substitutions within the technical scope disclosed in the present disclosure, which should all fall within the scope of protection of the present disclosure. Therefore, the scope of protection of the present disclosure shall be determined by the scope of protection of the claims.

## CLAIMS

1. An image scanning method, comprising:

obtaining a file to be detected in an image file;

splitting the file to be detected into a plurality of batches, and generating a scanning sub-task corresponding to each batch;

for the scanning sub-task of each batch, triggering a computing unit to load a scanning rule corresponding to the scanning sub-task, and performing, according to the scanning rule, computation corresponding to the scanning sub-task, wherein the computing unit is implemented through a Function Compute (FC) service; and

merging computation results corresponding to the scanning sub-tasks to obtain an image scanning result.

2. The method of claim 1, wherein the obtaining the file to be detected in the image file comprises: obtaining the image file from an image repository; and obtaining a pre-filtering rule, and filtering the image file based on the pre-filtering rule to determine the file to be detected.

3. The method of claim 1 or 2, wherein after the obtaining the file to be detected in the image file, the method further comprises transmitting the file to be detected to an intermediate storage space other than a service server.

4. The method of any one of claims 1 to 3, wherein the splitting the file to be detected into the plurality of batches and generating the scanning sub-task corresponding to each batch comprises: splitting the file to be detected into the plurality of batches according to a pre-configured concurrency strategy, and generating the scanning sub-task corresponding to each batch.

5. The method of any one of claims 1 to 4, wherein for the scanning sub-task of each batch, the triggering the computing unit to load the scanning rule corresponding to the scanning sub-task and performing, according to the scanning rule, the computation corresponding to the scanning sub-task comprises:

for the scanning sub-task of each batch, invoking an input component of the computing unit to trigger a computing execution component of the computing unit, loading, by the computing execution component, the scanning rule corresponding to the scanning sub-task, performing, according to the scanning rule, the computation corresponding to the scanning sub-task, and outputting, by an output component of the computing unit, the computation result corresponding to the scanning sub-task.

6. The method of any one of claims 1 to 5, wherein the merging the computation results corresponding to the scanning sub-tasks to obtain the image scanning result comprises:

distributing the computation results corresponding to respective scanning sub-tasks to recheck engines corresponding to the scanning sub-tasks, and adding an asynchronous-to-synchronous flag to some or all of the scanning sub-tasks; and

after obtaining recheck results corresponding to all scanning sub-tasks to which the asynchronous-to-synchronous flag is added, merging respective recheck results to obtain the image scanning result.

7. The method of claim 6, wherein the recheck result comprises a detection item name, a detection result, and a detection duration corresponding to the scanning sub-task.

8. An image scanning apparatus, comprising:

an obtaining module, configured to obtain a file to be detected in an image file;

a splitting module, configured to split the file to be detected into a plurality of batches and generate a scanning sub-task corresponding to each batch;

a computing module, configured to, for the scanning sub-task of each batch, trigger a computing unit to load a scanning rule corresponding to the scanning sub-task, and perform, according to the scanning rule, computation corresponding to the scanning sub-task, wherein the computing unit is implemented through a Function Compute (FC) service; and

a synchronization module, configured to merge computation results corresponding to the scanning sub-tasks to obtain an image scanning result.

9. The image scanning apparatus of claim 8, wherein the obtaining module is specifically configured to: obtain the image file from an image repository; and obtain a pre-filtering rule, and filter the image file based on the pre-filtering rule to determine the file to be detected.

10. The image scanning apparatus of claim 8 or 9, wherein after obtaining the file to be detected in the image file, the image scanning apparatus is further configured to transmit the file to be detected to an intermediate storage space other than a service server.

11. The image scanning apparatus of any one of claims 8 to 10, wherein the splitting module is specifically configured to: split the file to be detected into the plurality of batches according to a pre-configured concurrency strategy, and generate the scanning sub-task corresponding to each batch.

12. The image scanning apparatus of any one of claims 8 to 11, wherein the computing module is specifically configured to:

for the scanning sub-task of each batch, invoke an input component of the computing unit to trigger a computing execution component of the computing unit, load, by the computing execution component, the scanning rule corresponding to the scanning sub-task, perform, according to the scanning rule, the computation corresponding to the scanning sub-task, and output, by an output component of the computing unit, the computation result corresponding to the scanning sub-task.

13. An electronic device, comprising: a memory and a processor,

wherein the memory is configured to store a computer program;

the processor is configured to execute the computer program stored in the memory, and when the computer program is executed, the processor is caused to perform the method of any one of claims 1 to 7.

14. A computer-readable storage medium, wherein a computer program is stored in the computer-readable storage medium, and when the computer program is executed by a processor, the processor is caused to perform the method of any one of claims 1 to 7.

15. A computer program product comprising a computer program, wherein when the computer program is executed by a processor, the method of any one of claims 1 to 7 is implemented.