TEST METHOD AND SYSTEM FOR TARGET SYSTEM, AND STORAGE MEDIUM

TECHNICAL FIELD

[0001] This specification relates to the field of internet technologies, and in particular, to a test method and system for a target system, and a storage medium.

BACKGROUND

[0002] A target system usually needs to be tested when changes such as version update, online upgrade, and bug fix are made to the target system. An objective of testing is to determine that no risk is introduced in the current change and ensure that a target system after the change can run stably.

[0003] Usually, user acceptance testing (UAT) is mainly used to test the target system. Specifically, a large amount of real-time traffic is input to the target system after the change, and an acceptance person observes whether a response of the target system to the real-time traffic meets an expectation, and then determines, based on an observation result, whether a risk is introduced in the current change. However, a test result in UAT depends on visual observation of the acceptance person, which is time-consuming and labor-intensive. In addition, the test result is affected by a professional capability of the acceptance person and is not high in accuracy.

[0004] Content in the background is merely information known to the inventor, and neither means that the information has entered the public domain before the application date of this disclosure, nor means that the information can become the existing technology of this disclosure.

SUMMARY

[0005] This specification provides a test method and system for a target system, and a storage medium, to improve test efficiency and improve accuracy of a test result.

[0006] According to a first aspect, this specification provides a test method for a target system, including: receiving a test request corresponding to the target system, where the test request is used

to request to test whether there is a risk in a service of the target system for a target scenario after a target change occurs; obtaining traffic data corresponding to the target scenario based on the test request, and determining a first program corresponding to the target system before the target change occurs and a second program corresponding to the target system after the target change occurs; separately processing the traffic data by using the first program and the second program, to obtain first test data and second test data, where the first test data is intermediate data collected in a process of processing the traffic data by the first program, and the second test data is intermediate data collected in a process of processing the traffic data by the second program; and determining, based on the first test data and the second test data, whether there is a risk in the service of the target system for the target scenario after the target change occurs.

[0007] According to a second aspect, this specification further provides a test system for a target system, including: at least one storage medium that stores at least one instruction set for testing the target system; and at least one processor that is communicatively connected to the at least one storage medium. When the test system for a target system runs, the at least one processor reads the at least one instruction set, and performs the following operations based on an indication of the at least one instruction set:

receiving a test request corresponding to the target system, where the test request is used to request to test whether there is a risk in a service of the target system for a target scenario after a target change occurs; obtaining traffic data corresponding to the target scenario based on the test request, and determining a first program corresponding to the target system before the target change occurs and a second program corresponding to the target system after the target change occurs; separately processing the traffic data by using the first program and the second program, to obtain first test data and second test data, where the first test data is intermediate data collected in a process of processing the traffic data by the first program, and the second test data is intermediate data collected in a process of processing the traffic data by the second program; and determining, based on the first test data and the second test data, whether there is a risk in the service of the target system for the target scenario after the target change occurs.

[0008] According to a third aspect, this specification further provides a computer-readable nonvolatile storage medium. The computer-readable nonvolatile storage medium stores at least one instruction set, and when the at least one instruction set is executed by at least one processor, the test method for a target system provided in the first aspect is implemented.

[0009] Other functions of the test method and system for a target system, and the storage

medium provided in this specification are partially listed in the following descriptions. Creative aspects of the test method and system for a target system, and the storage medium provided in this specification can be fully explained by practice or by using the methods, apparatuses, and combinations described in the following detailed examples.

BRIEF DESCRIPTION OF DRAWINGS

[0010] To describe the technical solutions in the embodiments of this specification more clearly, the following briefly describes the accompanying drawings required for describing the embodiments. Clearly, the accompanying drawings in the following description show merely some embodiments of this specification, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0011] FIG. 1 is a schematic diagram of an application scenario of a test method for a target system according to an embodiment of this specification;

[0012] FIG. 2 is a diagram of a hardware structure of a computing device 200 according to an embodiment of this specification;

[0013] FIG. 3 is a flowchart of a test method for a target system according to an embodiment of this specification;

[0014] FIG. 4 is a schematic diagram of a source of traffic data in a first database according to an embodiment of this specification;

[0015] FIG. 5 is a schematic diagram of a test process for a target system according to an embodiment of this specification; and

[0016] FIG. 6 is a schematic diagram of a process of comparing first test data and second test data according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0017] Specific application scenarios and requirements of this specification are provided in the following descriptions, so that a person skilled in the art can manufacture and use content of this specification. Various local modifications to the disclosed embodiments are clear to a person skilled in the art. In addition, the general principles defined herein can be applied to other embodiments and applications without departing from the spirit and scope of this specification.

Therefore, this specification is not limited to the shown embodiments, but has a widest scope consistent with that of the claims.

[0018] The terms used herein are merely intended to describe specific example embodiments, and impose no limitation. For example, unless otherwise explicitly stated in the context, the singular forms "one", "an", and "the" used herein can include plural forms. When being used in this specification, the terms "include", "comprise", and/or "have" mean existence of an associated integer, step, operation, element, and/or component, but does not preclude existence of one or more other features, integers, steps, operations, elements, components, and/or groups or addition of other features, integers, steps, operations, elements, components, and/or groups to the system/method.

[0019] In consideration of the following descriptions, these and other features of this specification, operations and functions of related components of the structure, and economy of combination and manufacturing of components can be significantly improved. With reference to the accompanying drawings, all of these form a part of this specification. However, it should be clearly understood that the accompanying drawings are merely used for the purpose of illustration and description, and are not intended to limit the scope of this specification. It should be further understood that the accompanying drawings are not drawn to scale.

[0020] The flowcharts used in this specification show operations implemented by a system according to some embodiments of this specification. It should be clearly understood that the operations in the flowcharts may not be sequentially implemented. On the contrary, the operations can be implemented in a reverse sequence or simultaneously. In addition, one or more other operations can be added to the flowcharts, and one or more operations can be removed from the flowcharts.

[0021] For ease of description, the terms that appear below in this specification are first explained.

[0022] Term 1 is "scenario". In this specification, a target system can provide a service in a plurality of scenarios. One scenario can correspond to one function or one service of the target system. That is, a function/service provided by the target system corresponds to one service scenario (briefly referred to as a scenario). If the target system can provide 400 functions, the target system can correspond to 400 scenarios. For example, a scenario A can be a scenario in which the target system provides a transaction service, and a scenario B can be a scenario in which the target system provides a query service.

[0023] Term 2 is "low-traffic scenario". The low-traffic scenario is a scenario in which a data traffic volume is relatively small in a target system. For example, in some low-traffic scenarios, monthly average traffic is less than 30 pieces, or there are dozens of pieces of traffic only in a specific time period in a year. A person skilled in the art can understand that if scenarios are classified based on a traffic volume, service scenarios provided by the target system can be classified into a low-traffic scenario, a standard-traffic scenario, a high-traffic scenario, and the like. A traffic volume in the high-traffic scenario is greater than a traffic volume in the standard-traffic scenario, and the traffic volume in the standard-traffic scenario is greater than a traffic volume in the low-traffic scenario. It should be noted that a specific scenario classification manner is not limited in this specification.

[0024] Term 3 is "middleware". The middleware is a software component located between an application program and an operating system, and is configured to coordinate and connect different systems, services, or components, to implement data transmission, communication, and function expansion.

[0025] Term 4 is "gray list". The gray list refers to a user between a whitelist and a blacklist. This type of user does not exhibit behavior to be blacklisted. However, there is a specific risk in behavior of this type of user.

[0026] Term 5 is "offline data warehouse". When processing real-time data traffic in a target scenario, a target system can record the real-time data traffic, and store the recorded data traffic in a database based on the scenario. The stored data traffic in the database can be used for subsequent testing of the target system. In this specification, the database used to store the data traffic is referred to as the offline data warehouse.

[0027] Term 6 is "simple log service (SLS) middleware". The SLS middleware is a service that integrates functions such as log storage, query, and real-time analysis. The SLS middleware can implement centralized log storage and real-time log analysis, and features high scalability, high data security, and ease of integration. The SLS middleware is comprehensive, flexible, and easy-to-use log middleware. The SLS middleware can be used for a data pipeline in a test system.

[0028] Term 7 is "Flink". Flink is Apache Flink. Flink is an open source big data processing framework, and can perform distributed stream processing and batch processing. Flink is designed to process large-scale data, and can provide an accurate result and high performance. Flink is a commonly used big data processing tool, and is usually used for tasks such as log analysis, real-

time recommendation, and machine learning. Flink can be used for a comparison module in a test system.

[0029] An application scenario of this specification is described below.

[0030] This specification provides a test method for a target system. The method can be used to test whether there is a risk in a service of the target system for a target scenario after a target change occurs. A test system can obtain traffic data corresponding to the target scenario based on a test request corresponding to the target system, and determine, based on a status of processing the traffic data by a target system before the change and a status of processing the traffic data by a target system after the change, whether there is a risk in the service of the target system after the change for the target scenario.

[0031] It should be noted that the target system can be any service system, including but not limited to a payment system, a transaction system, a social system, a wallet system, a merchant system, or the like.

[0032] FIG. 1 is a schematic diagram of a scenario of testing a target system according to an embodiment of this specification. As shown in FIG. 1, the scenario 100 can include a test system 11 and a target system 12.

[0033] The target system 12 can be a system that provides various services. The target system 12 serves as a system under test. In some embodiments, the target system 12 can be a software system. In this case, the target system 12 can be deployed in a computing device, or deployed in a device cluster including a plurality of computing devices. In some embodiments, the target system 12 can be a hardware system or a system that combines software and hardware. The target system 12 can also be referred to as a service device, a server, a server cluster, or the like. The target system 12 can receive a service request from a user.

[0034] In this specification, the service request received by the target system 12 can be referred to as traffic of the target system 12. In some embodiments, when providing a service in response to the service request, the target system 12 can record real-time traffic (that is, a service request) for which a service is being provided, and store the real-time traffic in a database. During recording, the target system 12 can record a scenario corresponding to the real-time traffic based on a function called for the service request. During storage, the target system 12 can record the real-time traffic in the corresponding scenario. After the real-time traffic is stored in the database based on the scenario, the real-time traffic can be considered as historical traffic of the corresponding scenario.

In this way, historical traffic respectively corresponding to different scenarios is stored in the database. In this scenario, the database can be referred to as an offline data warehouse.

[0035] The test system 11 can perform testing on the target system in response to a test request. In this case, the test system 11 can store data or instructions for performing the test method for a target system described in this specification, and can execute or be configured to execute the data or the instructions. In some embodiments, the test system11 can include a hardware device with a data information processing function and a program required to drive the hardware device to work.

[0036] In some embodiments, the test system 11 can include a mobile device, a tablet computer, a laptop computer, a device built in a motor vehicle or similar content, or any combination thereof. In some embodiments, the mobile device can include a smart home device, a smart mobile device, a virtual reality device, an augmented reality device or a similar device, or any combination thereof. In some embodiments, the smart home apparatus can include a smart television, a desktop computer, or any combination. In some embodiments, the smart mobile device can include a smartphone, a personal digital assistant, a game device, a navigation device, or any combination thereof. In some embodiments, the virtual reality device or the augmented reality device may include a virtual reality helmet, virtual reality glasses, a virtual reality patch, an augmented reality helmet, augmented reality glasses, an augmented reality patch or similar content, or any combination thereof. For example, the virtual reality device or the augmented reality device may include Google glasses, a head mounted display, or VR. In some embodiments, the apparatus built in the motor vehicle can include a vehicle-mounted computer, a vehicle-mounted television, and the like.

[0037] In some embodiments, the test system 11 can receive a test request for the target system 12. The test request is used to request to test whether there is a risk in a service of the target system 12 for a target scenario after a target change occurs. The test system 11 can perform the test method for a target system described in this specification in response to the test request.

[0038] In some embodiments, the test system 11 can obtain traffic data corresponding to the target scenario in response to the test request. The traffic data can include real-time traffic and/or historical traffic. For example, the test system 11 can obtain the traffic data corresponding to the target scenario in at least one of the following manners: (1) retrieving historical traffic corresponding to the target scenario from the database (for example, the offline data warehouse described above), and (2) obtaining real-time traffic of the target system 12 in the target scenario.

Further, the test system 11 can control a target system 12 before the change and a target system 12 after the change to separately process the traffic data, and determine, based on statuses of processing the traffic data by the target system 12 before the change and the target system 12 after the change, whether there is a risk in the service of the target system 12 for the target scenario after the change occurs.

[0039] The test system 11, the target system 12, and the database can be communicatively connected to each other through a network, and transmit information or data to each other through the network. In some embodiments, the network can be any type of wired or wireless network, or a combination thereof. For example, the network can include a cable network, a wired network, an optical fiber network, a telecommunications network, an intranet, the internet, a local area network (LAN), a wide area network (WAN), a wireless local area network (WLAN), a metropolitan area network (MAN), a public switched telephone network (PSTN), a Bluetooth network, a ZigBee network, a near field communication (NFC) network, or a similar network. In some embodiments, the network can include one or more network access points. For example, the network can include a wired or wireless network access point, for example, a base station or an internet switching point. Through the access point, one or more components in the test system 11, the target system 12, and the database can be connected to the network to exchange data or information.

[0040] It should be understood that a quantity of test systems 11 and a quantity of target systems 12 in FIG. 1 are merely examples. Based on an implementation requirement, there can be any quantity of test systems 11 and target systems 12. Alternatively, any quantity of databases can be included.

[0041] It should be noted that the test system 11 and the target system 12 can correspond to a same physical device, or can correspond to different physical devices. This is not limited in this specification.

[0042] FIG. 2 is a diagram of a hardware structure of a computing device 200 according to an embodiment of this specification. The computing device 200 can serve as the test system 11 in FIG. 1 to perform the test method for a target system described in this specification.

[0043] As shown in FIG. 2, the computing device 200 can include at least one storage medium 230 and at least one processor 220. In some embodiments, the computing device 200 can further include a communication port 250 and an internal communication bus 210. The computing device

200 can further include an I/O component 260.

[0044] The internal communication bus 210 can be connected to different system components. For example, the internal communication bus 210 can be connected to the storage medium 230, the processor 220, the communication port 250, the I/O component 260, and the like.

[0045] The I/O component 260 supports input/output between the computing device 200 and other components.

[0046] The communication port 250 is used for data communication between the computing device 200 and the outside. For example, the communication port 250 can be used for data communication between the computing device 200 and a network 140. The communication port 250 can be a wired communication port, or can be a wireless communication port.

[0047] The storage medium 230 can include a data storage apparatus. The data storage apparatus can be a non-transitory storage medium, or can be a transitory storage medium. For example, the data storage apparatus can include one or more of a disk 232, a read-only storage medium (ROM) 234, or a random access storage medium (RAM) 236. The storage medium 230 further includes at least one instruction set stored in the data storage apparatus. The instruction set can include computer program code. The computer program code can include a program, a routine, an object, a component, a data structure, a process, a module, and the like.

[0048] The at least one processor 220 can be communicatively connected to the at least one storage medium 230. When the computing device 200 runs, the at least one processor 220 reads the at least one instruction set, and performs, based on an indication of the at least one instruction set, the test method for a target system provided in this specification. The processor 220 can perform the steps included in the test method for a target system. The processor 220 can be in a form of one or more processors. In some embodiments, the processor 220 can include one or more hardware processors, for example, a microcontroller, a microprocessor, a reduced instruction set computer (RISC), an application-specific integrated circuit (ASIC), an application-specific instruction set processor (ASIP), a central processing unit (CPU), a graphics processing unit (GPU), a physical processing unit (PPU), a microcontroller unit, a digital signal processor (DSP), a field programmable gate array (FPGA), an advanced RISC machine (ARM), a programmable logic device (PLD), any circuit, processor, or the like that can perform one or more functions, or any combination thereof.

[0049] To merely describe a problem, only one processor 220 is shown in the computing

device 200 in the accompanying drawings. However, it should be noted that the computing device 200 in this specification can further include a plurality of processors. Therefore, the operations and/or method steps disclosed in this specification can be performed by one processor, or can be jointly performed by a plurality of processors. For example, if the processor 220 in the computing device 200 described in this specification performs step A and step B, it should be understood as that step A and step B can be jointly or separately performed by two different processors 220 (for example, a first processor performs step A, and a second processor performs step B, or a first processor and a second processor jointly perform step A and step B).

[0050] FIG. 3 is a flowchart of a test method for a target system according to an embodiment of this specification. As described above, the test system 11 in FIG. 1 and the computing device 200 in FIG. 2 can perform the test method for a target system in this specification. Descriptions are provided below by using an example in which the test system 11 serves as an execution body. As shown in FIG. 3, the test method for a target system can include the following steps.

[0051] S310: Receive a test request corresponding to the target system, where the test request is used to request to test whether there is a risk in a service of the target system for a target scenario after a target change occurs.

[0052] The target change can be any change occurring on the target system 12, including but not limited to a code change, a configuration change, or the like. In an example, the target change occurring on the target system can be an upgrade to the target system, bug fix, version rollback, a configuration change, or the like.

[0053] The test request is used to indicate the test system 11 to test a service of the target system 12 in the target scenario after the target change occurs. Alternatively, the test request is used to request to test whether there is a risk in the service of the target system 12 for the target scenario after the target change occurs.

[0054] In some embodiments, with reference to the scenario shown in FIG. 1, the test request can be from the target system 12. For example, when detecting that the target change occurs on the target system 12, the target system 12 sends the test request to the test system 11. For another example, when detecting a version update instruction for the target system, the target system 12 can send the test request to the test system 11. For another example, when receiving a configuration change instruction for the target system, the target system 12 can send the test request to the test system 11. In the above case, in S310, the test system 11 can receive the test request from the target

system 12.

[0055] In some embodiments, the test request can alternatively be from a tester. For example, the test system 11 can provide an interaction interface. When the tester needs to test the target system 12, a test triggering operation can be input to the test system 11 through the interaction interface, for example, the to-be-tested target system, the target change, and the target scenario are specified in the interaction interface. In this case, in S310, when the test system 11 detects the test triggering operation of the tester, it is equivalent to receiving the test request for the target system.

[0056] S320: Obtain traffic data corresponding to the target scenario based on the test request, and determine a first program corresponding to the target system before the target change occurs and a second program corresponding to the target system after the target change occurs.

[0057] For ease of description, a program corresponding to the target system before the target change occurs is referred to as the first program, and a program corresponding to the target system after the target change occurs is referred to as the second program. That is, the first program is a program used to implement a service capability of the target system before the change, and the second program is a program used to implement a service capability of the target system after the change.

[0058] In some embodiments, the test request can include an identifier of the first program and an identifier of the second program. The identifier of the first program and the identifier of the second program can be version numbers of the programs, MD5 (Message-Digest Algorithm) codes, program names, program storage paths, or the like. In this case, the test system 11 can determine the first program based on the identifier of the first program, and determine the second program based on the identifier of the second program.

[0059] In some embodiments, the test request can include an identifier of the target change. In this case, the test system 11 can obtain a preset correspondence, and the preset correspondence represents identifiers of programs before changes and identifiers of programs after changes respectively corresponding to different changes. Further, the test system 11 can obtain an identifier of the first program and an identifier of the second program by querying the preset correspondence based on the identifier of the target change, and then determine the first program based on the identifier of the first program, and determine the second program based on the identifier of the second program.

[0060] In some embodiments, the test request can include an identifier of the second program.

In this case, the test system 11 can determine a program that is currently running on the target system 12 as the first program, and determine the second program based on the identifier of the second program.

[0061] In an example, the target scenario can be a low-traffic scenario, a standard-traffic scenario, a high-traffic scenario, or the like.

[0062] In some embodiments, the test request can include a type of the target scenario. The test system 11 can obtain the traffic data corresponding to the target scenario based on the type of the target scenario. For example, if the type of the target scenario is the low-traffic scenario, traffic data corresponding to the low-traffic scenario is obtained; if the type of the target scenario is the standard-traffic scenario, traffic data corresponding to the standard-traffic scenario is obtained; or if the type of the target scenario is the high-traffic scenario, traffic data corresponding to the high-traffic scenario is obtained. In some embodiments, the test request can further include a quantity of pieces of traffic data. For example, if there are 1000 pieces of traffic data, the test system 11 can obtain 1000 pieces of traffic data corresponding to the target scenario.

[0063] In some embodiments, when the target scenario is a low-traffic scenario, the test system 11 can obtain the traffic data corresponding to the target scenario in the following manner: obtaining the traffic data corresponding to the target scenario from a first database, where the first database is used to store traffic data respectively corresponding to at least one scenario.

[0064] The first database can also be referred to as a low-traffic warehouse, and includes the traffic data corresponding to the target scenario. FIG. 4 is a schematic diagram of a source of traffic data in a first database according to an embodiment of this specification. With reference to FIG. 4, the traffic data corresponding to the target scenario in the first database can be obtained in at least one of the following manners: Manner 1: Real-time traffic data generated by the target system 12 in the target scenario is recorded into the first database. Manner 2: Historical traffic data corresponding to the target scenario is obtained from a second database, and is written into the first database. The second database includes historical traffic data respectively corresponding to a plurality of scenarios. In this embodiment, the second database can be an offline data warehouse.

[0065] In some embodiments, for the low-traffic scenario, the first database can preferentially store new traffic in the low-traffic scenario. That is, a percentage of new traffic in the first database can be greater than or equal to a preset threshold. For example, the new traffic can be real-time traffic from the manner 1. If the quantity of pieces of traffic data specified in the test request is

1000, and the preset threshold is 5%, at least 50 pieces of traffic in the first database are from the manner 1, and the remaining traffic is from the manner 2. In this case, the traffic in the first database can be obtained in the following manner: At least 50 pieces of real-time traffic in the low-traffic scenario can be first recorded into the first database, and then 950 pieces of historical traffic in the low-traffic scenario can be pulled from the second database and written into the first database. That is, the first database includes 50 pieces of new traffic (real-time traffic) and 950 pieces of historical traffic, totaling 1000 pieces of traffic data. The traffic data is the traffic data corresponding to the target scenario.

[0066] In some embodiments, when a quantity of pieces of recorded real-time traffic (traffic from the manner 1) cannot meet the quantity requirement (for example, 50), the test system 11 can obtain historical traffic that is newer in time from the second database as new traffic. For example, historical traffic in a last year can be pulled as new traffic. When a quantity of pieces of historical traffic in the last year is still insufficient, traffic in past two years, past three years, or the like can be pulled until a sufficient quantity of pieces of new traffic is pulled.

[0067] In some embodiments, when the target scenario is a standard-traffic scenario, a high-traffic scenario, or the like, the test system 11 can directly record a sufficient quantity of pieces of real-time traffic into the first database. For example, if the quantity of pieces of traffic data specified in the test request is 1000, 1000 pieces of real-time traffic can be recorded into the first database, and the traffic data is the traffic data corresponding to the target scenario.

[0068] When testing is performed by using the real-time traffic, a more accurate test result usually can be obtained. However, for the low-traffic scenario, there is an excessively small quantity of pieces of real-time traffic or there is no real-time traffic. Therefore, in the above asynchronous traffic recording solution, the real-time traffic is recorded, and the historical traffic is pulled, to ensure that there is a sufficient quantity of pieces of traffic data for testing, and ensure that there is a specific quantity of pieces of real-time traffic. This further improves test accuracy while the low-traffic scenario can be tested.

[0069] Because a quantity of pieces of real-time traffic in each low-traffic scenario varies, the percentage of new traffic is adjusted based on a preset threshold for each low-traffic scenario, to implement customized testing for the low-traffic scenario. For a low-traffic scenario with excessively low real-time traffic, the percentage of new traffic can be decreased. For a low-traffic scenario with relatively low real-time traffic, the percentage of new traffic can be increased. In this

way, the percentage of new traffic can be accurately controlled without affecting testing of the low-traffic scenario, and test accuracy and test efficiency can be balanced, to further improve the test effect.

[0070] S330: Separately process the traffic data by using the first program and the second program, to obtain first test data and second test data, where the first test data is intermediate data collected in a process of processing the traffic data by the first program, and the second test data is intermediate data collected in a process of processing the traffic data by the second program.

[0071] Specifically, the test system 11 can control a target system before the change and a target system after the change to separately process the traffic data (or control the first program and the second program to separately process the traffic data), to obtain the first test data collected in a process of processing the traffic data by the target system before the change, and obtain the second test data collected in a process of processing the traffic data by the target system after the change.

[0072] For ease of understanding, detailed descriptions are provided below with reference to FIG. 5.

[0073] FIG. 5 is a schematic diagram of a test process for a target system according to an embodiment of this specification. As shown in FIG. 5, the test system 11 can include a control module. The control module is configured to control the target system before the change and the target system after the change. For example, the control module can obtain the traffic data corresponding to the target scenario from the first database. Further, the control module can input the traffic data to the target system (that is, the first program) before the change to trigger the first program to process the traffic data, and when the first program proceeds to a plurality of data collection points, perform data collection to obtain first collected data corresponding to the plurality of data collection points, and obtain the first test data based on the first collected data corresponding to the plurality of data collection points. That is, the first test data can include the first collected data respectively corresponding to the plurality of data collection points.

[0074] The data collection point is a preset data collection location in a program, and can also be referred to as a tracking point. When the program proceeds to a data collection point, information collection can be performed, to record related information when the program proceeds to the data collection point. In some embodiments, the plurality of data collection points can be preset fixed data collection points.

[0075] In some embodiments, the plurality of data collection points can be dynamically configured data collection points. For example, the test request can further include test customization information. The test system 11 can determine the plurality of data collection points based on the test customization information. In an example, the test customization information can include information about a data collection point, for example, a location of the data collection point, a data collection type, and a data collection rule. The test system 11 can determine the plurality of data collection points by using the information about the data collection point, and when the first program processes the traffic data and the first program proceeds to the plurality of data collection points, perform data collection to obtain the first collected data.

[0076] In some embodiments, the plurality of data collection points can include one or more of an input parameter or an output parameter of middleware called by the first program, and an original input, an attribute, a policy, a feature, a post-task, and a review task of the target scenario.

[0077] In an example, the middleware can use a basic service (function) provided by an operating system to connect parts or different applications in the target system to implement function sharing. The middleware called by the first program can include communication middleware, message middleware, and data middleware. For example, the communication middleware can be a remote procedure call (RPC). In this example, the data collection point can be set to collect an input parameter or an output parameter of the RPC.

[0078] In some embodiments, the message middleware can include a rocket message queue (RocketMQ), a message broker (MsgBroker), or the like. In this example, the data collection point can be set to collect an input parameter or output parameter of RocketMQ or MsgBroker.

[0079] In some embodiments, the data middleware can include a my structured query language (MySQL), a remote dictionary server (Redis), Elasticsearch (ES), a Hadoop database (HBase), and the like. In this example, the data collection point can be set to collect an input parameter or output parameter of MySQL, Redis, ES, or HBase.

[0080] In some embodiments, the original input of the target scenario can include an executor identifier (partenerId), an initiator identifier (tenantId), or an event code. The executor identifier can be used to indicate a party that provides a service in the target system, for example, a merchant in a transaction service. The initiator identifier can be used to indicate a party that requests to use a service in the target system, for example, a user in the transaction service. The event code can be used to indicate a service type provided by the target system. In this example, the data collection

point can be set to collect one or more of original inputs of the target scenario.

[0081] In some embodiments, the attribute of the target scenario can include address information, user information, card information, transaction order information, device information, or the like. For example, for the transaction service, the address information can be a delivery address provided by an initiator; the user information can be a name of an executor or the initiator (for example, a name or a company name); the card information can be a tail number of a bank card, a tail number of a payment account, or the like that is used when the initiator pays; the transaction order information can be an order number of a transaction order, a logistics document number corresponding to an order, or the like; and the device information can include an identifier of a device used by the initiator. In this example, the data collection point can be set to collect one or more of attributes of the target scenario.

[0082] In some embodiments, the feature of the target scenario can include a blacklist, a whitelist, a gray list, and historical data corresponding to the target scenario, and authenticity of an address in the attribute. The historical data can include a historical transaction amount, a transaction frequency, and the like. In this example, the data collection point can be set to collect one or more of features of the target scenario.

[0083] In some embodiments, the policy of the target scenario can be a risk control policy corresponding to the target scenario. In an example, the risk control policy can be that a calculated value is obtained based on the feature of the target scenario, and it is determined, based on the calculated value, whether there is a risk. For example, if the historical transaction amount is used for the feature of the target scenario, the historical transaction amount can be calculated by using a preset feature algorithm, and a calculated value can be obtained. When the calculated value is greater than a threshold, it is determined that there is a risk in the target scenario. In this example, the data collection point can be set to collect a risk control result obtained based on the policy. For example, the data collection point can be used to collect information indicating that there is a risk, information indicating that there is no risk, information indicating transfer to manual review, or the like.

[0084] In some embodiments, the post-task of the target scenario can be to store risk control data of the target scenario in a case of current traffic. In this example, the data collection point can be set to collect information indicating whether storage is performed and a storage result (whether successful storage is implemented).

[0085] In some embodiments, the review task of the target scenario can be a manual review task initiated when the risk control result obtained based on the policy is "transfer to manual review". In this example, the data collection point can be set to collect information indicating whether the manual review task is initiated and a result of the manual review task (whether there is a risk).

[0086] In some embodiments, the first program can sequentially process traffic data in the first database. Each time one piece of traffic data is processed, the test system 11 can collect a group of first collected data based on the plurality of data collection points. With reference to FIG. 4, after the first program completes processing of the traffic data in the first database, the test system can process a plurality of groups of first collected data to obtain the first test data. For example, the first collected data can be preprocessed and packaged to obtain the first test data.

[0087] In the above embodiment, the plurality of data collection points are determined based on the test customization information, and the test data is collected at the plurality of data collection points, to implement customized data collection. Therefore, the collected test data is more targeted, thereby improving accuracy of the test method for a target system. In addition, customized data collection can further make the test method for a target system adapt to a plurality of different target systems and target scenarios. For example, different data collection points can be customized for different target systems or target scenarios, to collect required test data in a targeted manner. This improves applicability of the test method.

[0088] A process of controlling the target system (the first program) before the change by the test system 11 is described above. Similarly, the test system 11 can further input the traffic data to the target system (the second program) after the change to trigger the second program to process the traffic data, and when the second program proceeds to the plurality of data collection points, perform data collection to obtain second collected data corresponding to the plurality of data collection points, and obtain the second test data based on the second collected data corresponding to the plurality of data collection points.

[0089] It should be noted that for a process of controlling the second program by the test system 11, refer to the above related descriptions of the process of controlling the first program by the test system 11. Details are not described herein. It should be noted that the test system 11 can simultaneously perform the processes of controlling the first program and the second program, or can sequentially perform the processes of controlling the first program and the second program.

The test system 11 can control the first program and the second program in any sequence. This is not limited in this specification.

[0090] In some embodiments, the test system 11 can collect the first test data and the second test data by using a plug-in program based on a bytecode injection technology. The plug-in program based on the bytecode injection technology can insert bytecode when the target system 12 runs, to collect data or input data without modifying source code of the target system. In an example, the plug-in program can be implemented based on a plug-in or a library such as an ASMifier class visitor (ASM), a code generation library (CGLIB), a byte buddy, or Java bytecode (Javassist), or can be customized based on an actual application situation. An implementation of the plug-in program is not limited in this application.

[0091] In some embodiments, the plug-in program can be deployed in the target system. When the plug-in program is deployed in the target system, the test system 11 can further determine the plurality of data collection points based on the test customization information in the test request. Further, the test system 11 provides the plurality of data collection points to the plug-in program. After the test system 11 inputs the traffic data to the first program, when the first program processes the traffic data, the plug-in program collects the first collected data respectively corresponding to the plurality of received data collection points based on the data collection points. After the test system 11 inputs the traffic data to the second program, when the second program processes the traffic data, the plug-in program collects the second collected data respectively corresponding to the plurality of received data collection points based on the data collection points. It should be understood that the plurality of data collection points corresponding to the first program are the same as the plurality of data collection points corresponding to the second program.

[0092] In some embodiments, when the test system 11 and the target system 12 are deployed on a same hardware entity, the plug-in program runs on the same hardware entity. When the test system 11 is deployed on a test device, and the target system is deployed on a service device, the plug-in program can be deployed on the service device, and exchange data with the test system 11 through a communication link between the test device and the service device.

[0093] In this specification, the plug-in based on the bytecode injection technology is used, and data collection can be performed at the plurality of data collection points without changing the first program or the second program, so that data can be collected without intruding into the source code, to ensure stability of the target system in a test process. In addition, the plug-in based on the

bytecode injection technology can be deployed or removed at any time. By using the plug-in, the test method for a target system provided in this specification can be dynamically pluggable.

[0094] S340: Determine, based on the first test data and the second test data, whether there is a risk in the service of the target system for the target scenario after the target change occurs.

[0095] The test system can compare the first test data and the second test data, to determine whether there is a risk in the service of the target system for the target scenario after the target change occurs. For example, the first test data includes the first collected data corresponding to the plurality of data collection points, and the second test data includes the second collected data corresponding to the plurality of data collection points. Therefore, the test system can perform data comparison between collected data corresponding to a same data collection point in the first test data and the second test data, to determine whether there is a risk in the service of the target system for the target scenario after the target change occurs.

[0096] Still with reference to FIG. 5, in some embodiments, the test system 11 can further include a data pipeline and a comparison module. After obtaining the first test data and the second test data, the control module can send the first test data and the second test data to the comparison module through the data pipeline. The data pipeline can be implemented by using a capability of an SLS log middleware. The comparison module can perform data comparison between the first test data and the second test data, to obtain a test result. The comparison module can be implemented by using a capability of Blink middleware. Blink is an optimized and improved version of Flink, and can meet a requirement of processing a large amount of data. Alternatively, a big data component such as an offline data warehouse task and Spark is used for implementation. This is not limited in this application.

[0097] In some embodiments, the test system 11 can obtain a data comparison solution corresponding to the target scenario from a third database, and then perform data comparison between the first test data and the second test data based on the data comparison solution, to determine whether there is a risk in the service of the target system for the target scenario after the target change occurs.

[0098] In some embodiments, the third database can also be referred to as a rule warehouse, and the third database can store a plurality of comparison rules. One of the comparison rules represents a comparison manner for two data objects. For example, a comparison success is considered if a specific condition is met, and a comparison failure is considered if a specific

condition is met. For example, the comparison rules stored in the third database can include but are not limited to the following several types:

[0099] Comparison rule 1: For tree data (for example, an execution path of traffic), it is compared whether paths of the tree data are the same.

[0100] Comparison rule 2: For data of a string type, it is compared whether two strings are equal.

[0101] Comparison rule 3: For numeric data, two values are first converted based on configuration precision, and then it is compared whether two values obtained after conversion are equal.

[0102] Comparison rule 4: This is an ignore rule. That is, it is considered, by default, that two data objects are equal, that is, a comparison success is considered by default.

[0103] Comparison rule 5: This is a range rule in which it is identified whether two data objects fall within a same range. If both the two data objects fall within the same range, there is a comparison success. For example, an initiator identifier is used as an example. If both an initiator identifier collected in a running process of the first program and an initiator identifier collected in a running process of the second program fall within a blacklist or a whitelist, it is considered that the two data objects are equal, that is, there is a comparison success.

[0104] Comparison rule 6: This is a pattern matching rule. For example, a regular expression can be used to match a string pattern. The string pattern is also referred to as a text pattern and refers to a rule or template that can be used to match a class of strings. The regular expression is a language used to describe the string pattern. In an example, using the regular expression to match the string pattern means using the regular expression to find a string that conforms to a specific rule or template. When a same string can be matched between two data objects by using a same regular expression, it is determined that the two data objects are equal, that is, there is a comparison success.

[0105] Comparison rule 7: This is a null value comparison rule. Because a comparison result that is "true" or "false" cannot be obtained when comparison is performed between a null value and any value (including a null value), an obtained result is unknown. Therefore, when there is a null value in two data objects, special processing can be performed to ensure that a comparison result can be obtained. For example, "IS NULL" or "IS NOT NULL" can be used to check whether the two data objects are null values. When check results of both the two data objects are "value IS

NULL", it is determined that the two data objects are equal (if one of the two data objects is not null, a check result of the value is "value IS NOT NULL"), that is, there is a comparison success.

[0106] The third database can further store data comparison solutions corresponding to different scenarios. A data comparison solution corresponding to one scenario includes a mapping relationship between the plurality of data collection points and the plurality of comparison rules. One data collection point can correspond to one or more of the above comparison rules. Data comparison can be performed between collected data corresponding to a data collection point by using a comparison rule corresponding to the data collection point. It should be noted that specific content of the mapping relationship is not limited in this specification.

[0107] The third database stores data comparison solutions corresponding to different scenarios, and in a data comparison solution of a same scenario, different data collection points respectively correspond to comparison rules. Therefore, when comparing the first test data and the second test data, the test system 11 can compare data at different data collection points by using different comparison rules. It can be learned that the above solution implements customization of comparison rules in different scenarios, can adapt to more scenarios, and more accurately discovers a risk in the target system.

[0108] When comparing the first test data and the second test data, the test system 11 can traverse the plurality of data collection points, and for each current data collection point, perform data comparison between first collected data corresponding to the current data collection point and second collected data corresponding to the current collection point based on a comparison rule corresponding to the current data collection point in the data comparison solution, to obtain a comparison result corresponding to the current data collection point.

[0109] In some embodiments, one data collection point can correspond to a plurality of comparison rules, that is, data at the data collection point can be compared by using the plurality of comparison rules. For example, if data collected at a data collection point is an initiator identifier, a comparison rule corresponding to the data collection point can include the comparison rule 2 (to compare string content of the initiator identifier), the comparison rule 5 (to query whether an initiator corresponding to the initiator identifier is in the whitelist or the blacklist), the comparison rule 7 (to determine that the obtained initiator identifier is null), and the like. When one data collection point corresponds to a plurality of comparison rules, the test system 11 can separately perform data comparison based on the plurality of comparison rules corresponding to the data

collection point, to obtain a comparison result corresponding to the data collection point. For example, data comparison is performed between first collected data corresponding to the data collection point and second collected data corresponding to the data collection point based on the comparison rule 2, to obtain a comparison result corresponding to the comparison rule 2; data comparison is performed between the first collected data corresponding to the data collection point and the second collected data corresponding to the data collection point based on the comparison rule 5, to obtain a comparison result corresponding to the comparison rule 5; and data comparison is performed between the first collected data corresponding to the data collection point and the second collected data corresponding to the data collection point based on the comparison rule 7, to obtain a comparison result corresponding to the comparison rule 7.

[0110] Further, the test system 11 can comprehensively consider the comparison results respectively corresponding to the comparison rule 2, the comparison rule 5, and the comparison rule 7, to obtain a comparison result corresponding to the data collection point. For example, if all the comparison results corresponding to the comparison rule 2, the comparison rule 5, and the comparison rule 7 are a comparison success, the comparison result corresponding to the data collection point is a comparison success; or if a comparison result of at least one of the comparison rule 2, the comparison rule 5, and the comparison rule 7 is a comparison failure, the comparison result corresponding to the data collection point is a comparison failure.

[0111] FIG. 6 is a schematic diagram of a process of comparing first test data and second test data according to an embodiment of this specification. As shown in FIG. 6, five data collection points (A, B, C, D, and E) are used as examples. The first test data includes first collected data respectively corresponding to the data collection points A, B, C, D, and E, and the second test data includes second collected data respectively corresponding to the data collection points A, B, C, D, and E.

[0112] The test system 11 can compare the first collected data corresponding to the data collection point A and the second collected data corresponding to the data collection point A based on a comparison rule corresponding to the data collection point A, to obtain a comparison result corresponding to the data collection point A. The test system 11 can further compare the first collected data corresponding to the data collection point B and the second collected data corresponding to the data collection point B based on a comparison rule corresponding to the data collection point B, to obtain a comparison result corresponding to the data collection point B. By

analogy, the test system 11 can separately obtain a comparison result corresponding to the data collection point C, a comparison result corresponding to the data collection point D, and a comparison result corresponding to the data collection point E.

[0113] After obtaining the comparison results respectively corresponding to the plurality of data collection points, the test system 11 can determine, based on the comparison results respectively corresponding to the plurality of data collection points, whether there is a risk in the service of the target system for the target scenario after the target change occurs. For example, with reference to FIG. 6, the test system 11 can determine a test result based on the comparison results respectively corresponding to the data collection points A, B, C, D, and E, that is, whether there is a risk in the service of the target system for the target scenario after the target change occurs.

[0114] In some embodiments, if all the comparison results respectively corresponding to the plurality of data collection points are a comparison success, it is determined that there is no risk in the service of the target system for the target scenario after the target change occurs; or if a comparison result corresponding to at least one of the plurality of data collection points is a comparison failure, it is determined that there is a risk in the service of the target system for the target scenario after the target change occurs. In an example, with reference to FIG. 6, when all the comparison results of the five data collection points are a comparison success, it can be determined that there is no risk in the service of the target system for the target scenario after the target change occurs; or if there is a comparison failure in the comparison results of the five data collection points, it is determined that there is a risk in the service of the target system for the target scenario after the target change occurs.

[0115] In some other embodiments, the plurality of data collection points corresponding to the target scenario can be divided into two sets: a key collection point set and a non-key collection point set. The key collection point set includes some data collection points, and the non-key collection point set includes the other data collection points. If all comparison results respectively corresponding to the data collection points in the key collection point set are a comparison success, it is determined that there is no risk in the service of the target system for the target scenario after the target change occurs; or if a comparison result corresponding to at least one data collection point in the key collection point set is a comparison failure, it is determined that there is a risk in the service of the target system for the target scenario after the target change occurs.

[0116] In some other embodiments, on the basis of the above example, when all the comparison results respectively corresponding to the data collection points in the key collection point set are a comparison success, a comparison result of the data collection point in the non-key collection point set can be checked. When the comparison result of the data collection point in the non-key collection point set meets a condition, it is determined that there is no risk in the service of the target system for the target scenario after the target change occurs. Otherwise, it is determined that there is a risk in the service of the target system for the target scenario after the target change occurs.

[0117] In an example, that the comparison result of the data collection point in the non-key collection point set meets a condition can include: A percentage of data collection points with a comparison failure in the non-key collection point set is less than or equal to a preset percentage; or a quantity of data collection points with a comparison failure in the non-key collection point set is less than or equal to a preset quantity.

[0118] For example, the preset quantity can be 3. When all the comparison results respectively corresponding to the data collection points in the key collection point set are a comparison success, and the quantity of data collection points with a comparison failure in the non-key collection point set is less than or equal to 3, it is determined that there is no risk in the service of the target system for the target scenario after the target change occurs.

[0119] For example, it is assumed that a quantity of data collection points in the non-key collection point set is 50, and the preset percentage is 5% (rounding up). When all the comparison results respectively corresponding to the data collection points in the key collection point set are a comparison success, and the quantity of data collection points with a comparison failure in the non-key collection point set is less than or equal to 3 (50*5%=2.5, rounded up to 3), it is determined that there is no risk in the service of the target system for the target scenario after the target change occurs.

[0120] In some embodiments, when determining that there is a risk in the service of the target system 12 for the target scenario after the target change occurs, the test system 11 can generate and output alarm information.

[0121] In an example, the alarm information can indicate that there is a risk in the service of the target system 12 for the target scenario after the target change occurs. By generating and outputting the alarm information, the test system 11 can remind, in a timely manner, an operation

and maintenance person to process the risk to avoid a fault during online operation.

[0122] In some embodiments, the alarm information can further indicate a triggering location corresponding to the risk. For example, the triggering location can be a code location that causes the risk in the second program. For example, the test system 11 can determine the triggering location corresponding to the risk in the second program based on comparison between the first test data and the second test data. For example, the test system 11 can locate a data collection point with a comparison failure in the plurality of data collection points; and then use the data collection point as the triggering location corresponding to the risk, and write the triggering location into the alarm information. In the alarm information, the test system 11 can indicate the triggering location corresponding to the risk by using a name or a number of the data collection point or in another manner.

[0123] The test system 11 indicates the triggering location corresponding to the risk in the alarm information, to facilitate subsequent maintenance, so that the operation and maintenance person can more quickly process the risk, to improve maintenance efficiency.

[0124] In some embodiments, the test system 11 can further store a risk evaluation result for the target scenario in the target system in target storage space. When it is determined, through evaluation, that there is a risk, the test system 11 can further store the alarm information in the target storage space.

[0125] The test system 11 stores the risk evaluation result and the alarm information in the target storage space, to provide a reference for subsequent testing and maintenance, and further improve maintenance efficiency.

[0126] In conclusion, according to the test method and system for a target system, and the storage medium provided in this specification, after receiving the test request corresponding to the target system, the test system 11 can obtain the traffic data corresponding to the target scenario. Further, the test system 11 controls the first program (the program corresponding to the target system 12 before the change) and the second program (the program corresponding to the target system 12 after the change) to separately process the same traffic data, to obtain the first test data and the second test data; and determines, based on the first test data and the second test data, whether there is a risk in the service of the target system 12 for the target scenario after the target change occurs. In the above solution, statuses of processing the same traffic data by the target system 12 before the change and the target system 12 after the change are compared, to

automatically perform risk prevention and control testing on the target system 12 after the change. Compared with those in UAT, the test efficiency can be improved, manpower costs for testing can be lowered, and accuracy of the test result can be improved. Further, because the first test data is intermediate data collected in the process of processing the traffic data by the first program, and the second test data is intermediate data collected in the process of processing the traffic data by the second program, in the above solution, comparison between intermediate processes of executing the traffic data by the target system before the change and the target system after the change can be implemented, instead of simple comparison of execution results of the traffic data. Therefore, the accuracy of the test result can be further improved, and when the test result is that there is a risk, this helps trace a risk generation cause, and increases interpretability of the test result. In addition, the above solution can be applied to testing the target system in any scenario, and is widely applicable.

[0127] According to another aspect of this specification, a computer-readable non-transitory storage medium is provided and stores at least one instruction set for testing a target system. When the at least one instruction set is executed by a processor, the at least one instruction set instructs the processor to implement the steps of the test method for a target system described in this specification. In some possible implementations, various aspects of this specification can be further implemented in a form of a program product, including program code. When the program product runs on the computing device 200, the program code is used to enable the computing device 200 to perform the steps of the test method for a target system described in this specification. The program product configured to implement the above method can use a portable compact disc read-only memory (CD-ROM) that includes program code, and can run on the computing device 200. However, the program product in this specification is not limited thereto. In this specification, a readable storage medium can be any tangible medium that includes or stores a program, and the program can be used by or in combination with an instruction execution system. The program product can be any combination of one or more readable media. The readable medium can be a readable signal medium or a readable storage medium. For example, the readable storage medium can be but is not limited to an electrical system, apparatus, or device, a magnetic system, apparatus, or device, an optical system, apparatus, or device, an electromagnetic system, apparatus, or device, an infrared system, apparatus, or device, or a semiconductor system, apparatus, or device, or any combination thereof. More specific examples of the readable storage medium include an electrical

connection including one or more conducting wires, a portable disk, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM) or a flash memory, an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any proper combination thereof. The computer-readable storage medium can include a data signal propagated in a baseband or as part of a carrier, and carries readable program code. The propagated data signal can be in a plurality of forms, and includes but is not limited to an electromagnetic signal, an optical signal, or any proper combination thereof. The readable storage medium can alternatively be any readable medium other than the readable storage medium, and the readable medium can send, propagate, or transmit a program used by or in combination with an instruction execution system, apparatus, or device. The program code included in the readable storage medium can be transmitted by using any proper medium, including but not limited to a wireless medium, a wired medium, an optical cable, an RF medium, or any proper combination thereof. Program code for performing the operations in this specification can be written by using any combination of one or more program design languages. The program design language includes an object-oriented program design language, for example, Java or C++, and further includes a conventional procedural program design language, for example, "C" language, or a similar program design language. The program code can be completely executed on the computing device 200, partially executed on the computing device 200, executed as an independent software package, partially executed on the computing device 200 and partially executed on a remote computing device, or completely executed on a remote computing device.

[0128] Specific embodiments of this specification are described above. Other embodiments fall within the scope of the appended claims. In some cases, the actions or steps described in the claims can be performed in a sequence different from that in the embodiments and desired results can still be achieved. In addition, the process depicted in the accompanying drawings does not necessarily need a particular sequence or consecutive sequence to achieve the desired results. In some implementations, multitasking and parallel processing are possible or may be advantageous.

[0129] In conclusion, after reading the detailed disclosure, a person skilled in the art can understand that the detailed disclosure can be presented only by using an example, and can impose no limitation. Although it is not explicitly stated herein, a person skilled in the art can understand that the requirements of this specification include various proper changes, improvements, and

modifications to the embodiments. These changes, improvements, and modifications are intended to be provided in this specification, and fall within the spirit and scope of the example embodiments of this specification.

[0130] In addition, some terms in this specification are used to describe the embodiments of this specification. For example, "one embodiment", "embodiments", and/or "some embodiments" mean/means that specific features, structures, or characteristics described with reference to the embodiment can be included in at least one embodiment of this specification. Therefore, it can be emphasized and understood that two or more references to "embodiments", "one embodiment", or "alternative embodiments" in various parts of this specification do not necessarily refer to a same embodiment. In addition, specific features, structures, or characteristics can be properly combined in one or more embodiments of this specification.

[0131] It should be understood that in the above descriptions of the embodiments of this specification, to help understand a feature, for the purpose of simplifying this specification, various features are combined in a single embodiment, accompanying drawings, or descriptions thereof in this specification. However, this does not mean that the combination of these features is necessary. It is entirely possible for a person skilled in the art to label some devices and understand the devices as separate embodiments when reading this specification. That is, the embodiments of this specification can also be understood as an integration of a plurality of secondary embodiments. Content of each secondary embodiment is also true when a quantity of features is less than a quantity of all features in one of the above disclosed embodiments.

[0132] Each patent, patent application, publication of a patent application, and other materials, for example, articles, books, instructions, publications, documents, and products, that are cited in this specification and that are other than any document that is inconsistent or conflict with this document or that imposes limiting impact on the widest scope of the claims can be incorporated herein by reference and used for all purposes now or later associated with this document. In addition, the terms in this document shall prevail if there is any inconsistency or conflict between the descriptions, definitions, and/or use of the related terms in any material and the descriptions, definitions, and/or use of the related terms in this document.

[0133] Finally, it should be understood that the principles of the implementation solutions of this specification are described in the implementation solutions of the application disclosed in this specification. Other modified embodiments also fall within the scope of this specification.

Therefore, the embodiments disclosed in this specification are merely examples and not limitations. A person skilled in the art can use an alternative configuration according to the embodiments of this specification to implement the application in this specification. Therefore, the embodiments of this specification are not limited to the embodiments accurately described in the application.

CLAIMS

What is claimed is:

1. A test method for a target system, comprising:

receiving a test request corresponding to the target system, wherein the test request is used to request to test whether there is a risk in a service of the target system for a target scenario after a target change occurs;

obtaining traffic data corresponding to the target scenario based on the test request, and determining a first program corresponding to the target system before the target change occurs and a second program corresponding to the target system after the target change occurs;

separately processing the traffic data by using the first program and the second program, to obtain first test data and second test data, wherein the first test data is intermediate data collected in a process of processing the traffic data by the first program, and the second test data is intermediate data collected in a process of processing the traffic data by the second program; and

determining, based on the first test data and the second test data, whether there is a risk in the service of the target system for the target scenario after the target change occurs.

2. The method according to claim 1, wherein the separately processing the traffic data by using the first program and the second program, to obtain first test data and second test data comprises:

inputting the traffic data to the first program to trigger the first program to process the traffic data, and when the first program proceeds to a plurality of data collection points, performing data collection to obtain first collected data corresponding to the plurality of data collection points, and obtaining the first test data based on the first collected data corresponding to the plurality of data collection points; and

inputting the traffic data to the second program to trigger the second program to process the traffic data, and when the second program proceeds to the plurality of data collection points, performing data collection to obtain second collected data corresponding to the plurality of data collection points, and obtaining the second test data based on the second collected data corresponding to the plurality of data collection points.

3. The method according to claim 2, wherein the test request comprises test customization information, and the method further comprises:

determining the plurality of data collection points based on the test customization information.

4. The method according to claim 3, wherein the first collected data and the second collected data are collected by using a plug-in program based on a bytecode injection technology; and

the method further comprises: providing the plurality of data collection points to the plug-in program.

5. The method according to claim 1, wherein the target scenario is a low-traffic scenario, and the obtaining traffic data corresponding to the target scenario comprises:

obtaining the traffic data corresponding to the target scenario from a first database, wherein the first database is used to store traffic data respectively corresponding to at least one scenario.

6. The method according to claim 5, wherein the traffic data corresponding to the target scenario in the first database is obtained in at least one of the following manners:

recording real-time traffic data generated by the target system in the target scenario into the first database; or

obtaining historical traffic data corresponding to the target scenario from a second database, and writing the historical traffic data into the first database, wherein the second database comprises historical traffic data respectively corresponding to a plurality of scenarios.

7. The method according to claim 1, wherein the determining, based on the first test data and the second test data, whether there is a risk in the service of the target system for the target scenario after the target change occurs comprises:

obtaining a data comparison solution corresponding to the target scenario from a third database; and

performing data comparison between the first test data and the second test data based on the data comparison solution, to determine whether there is a risk in the service of the target system for the target scenario after the target change occurs.

8. The method according to claim 7, wherein the first test data comprises first collected data corresponding to a plurality of data collection points, the second test data comprises second collected data corresponding to the plurality of data collection points, and the data comparison solution comprises comparison rules respectively corresponding to the plurality of data collection points; and

the performing data comparison between the first test data and the second test data based on the data comparison solution, to determine whether there is a risk in the service of the target system

for the target scenario after the target change occurs comprises:

traversing the plurality of data collection points, and for each current data collection point, performing data comparison between first collected data corresponding to the current data collection point and second collected data corresponding to the current data collection point based on a comparison rule corresponding to the current data collection point in the data comparison solution, to obtain a comparison result corresponding to the current data collection point; and

determining, based on comparison results respectively corresponding to the plurality of data collection points, whether there is a risk in the service of the target system for the target scenario after the target change occurs.

9. The method according to claim 8, wherein the determining, based on comparison results respectively corresponding to the plurality of data collection points, whether there is a risk in the service of the target system for the target scenario after the target change occurs comprises:

if all the comparison results respectively corresponding to the plurality of data collection points are a comparison success, determining that there is no risk in the service of the target system for the target scenario after the target change occurs; or

if a comparison result corresponding to at least one of the plurality of data collection points is a comparison failure, determining that there is a risk in the service of the target system for the target scenario after the target change occurs.

10. The method according to claim 1, wherein after it is determined that there is a risk, the method further comprises:

generating and outputting alarm information, wherein the alarm information indicates that there is a risk in the service of the target system for the target scenario after the target change occurs.

11. The method according to claim 10, wherein the method further comprises:

determining a triggering location corresponding to the risk in the second program based on comparison between the first test data and the second test data, wherein

the alarm information further indicates the triggering location.

12. A test system for a target system, comprising:

at least one storage medium that stores at least one instruction set for testing the target system; and

at least one processor that is communicatively connected to the at least one storage medium,

wherein when the test system runs, the at least one processor reads the at least one instruction set, and performs the following operations based on an indication of the at least one instruction set:

receiving a test request corresponding to the target system, wherein the test request is used to request to test whether there is a risk in a service of the target system for a target scenario after a target change occurs;

obtaining traffic data corresponding to the target scenario based on the test request, and determining a first program corresponding to the target system before the target change occurs and a second program corresponding to the target system after the target change occurs;

separately processing the traffic data by using the first program and the second program, to obtain first test data and second test data, wherein the first test data is intermediate data collected in a process of processing the traffic data by the first program, and the second test data is intermediate data collected in a process of processing the traffic data by the second program; and

determining, based on the first test data and the second test data, whether there is a risk in the service of the target system for the target scenario after the target change occurs.

13. The test system according to claim 12, wherein to obtain the first test data and the second test data, the at least one processor performs the following operations:

inputting the traffic data to the first program to trigger the first program to process the traffic data, and when the first program proceeds to a plurality of data collection points, performing data collection to obtain first collected data corresponding to the plurality of data collection points, and obtaining the first test data based on the first collected data corresponding to the plurality of data collection points; and

inputting the traffic data to the second program to trigger the second program to process the traffic data, and when the second program proceeds to the plurality of data collection points, performing data collection to obtain second collected data corresponding to the plurality of data collection points, and obtaining the second test data based on the second collected data corresponding to the plurality of data collection points.

14. The test system according to claim 13, wherein the test request comprises test customization information, and the at least one processor further performs the following operation:

determining the plurality of data collection points based on the test customization information.

15. The test system according to claim 14, wherein the first collected data and the second collected data are collected by using a plug-in program based on a bytecode injection technology;

and the at least one processor further performs the following operation:

providing the plurality of data collection points to the plug-in program.

16. The test system according to claim 12, wherein the target scenario is a low-traffic scenario, and to obtain the traffic data corresponding to the target scenario, the at least one processor performs the following operation:

obtaining the traffic data corresponding to the target scenario from a first database, wherein the first database is used to store traffic data respectively corresponding to at least one scenario.

17. The test system according to claim 16, wherein the traffic data corresponding to the target scenario in the first database is obtained in at least one of the following manners:

recording real-time traffic data generated by the target system in the target scenario into the first database; or

obtaining historical traffic data corresponding to the target scenario from a second database, and writing the historical traffic data into the first database, wherein the second database comprises historical traffic data respectively corresponding to a plurality of scenarios.

18. The test system according to claim 12, wherein to determine whether there is a risk in the service of the target system for the target scenario after the target change occurs, the at least one processor performs the following operations:

obtaining a data comparison solution corresponding to the target scenario from a third database; and

performing data comparison between the first test data and the second test data based on the data comparison solution, to determine whether there is a risk in the service of the target system for the target scenario after the target change occurs.

19. The test system according to claim 18, wherein the first test data comprises first collected data corresponding to a plurality of data collection points, the second test data comprises second collected data corresponding to the plurality of data collection points, and the data comparison solution comprises comparison rules respectively corresponding to the plurality of data collection points; and

to determine whether there is a risk in the service of the target system for the target scenario after the target change occurs, the at least one processor performs the following operations:

traversing the plurality of data collection points, and for each current data collection point, performing data comparison between first collected data corresponding to the current data

collection point and second collected data corresponding to the current data collection point based on a comparison rule corresponding to the current data collection point in the data comparison solution, to obtain a comparison result corresponding to the current data collection point; and

determining, based on comparison results respectively corresponding to the plurality of data collection points, whether there is a risk in the service of the target system for the target scenario after the target change occurs.

20. A computer-readable nonvolatile storage medium, wherein the computer-readable nonvolatile storage medium stores at least one instruction set, and when the at least one instruction set is executed by at least one processor, the test method for a target system according to any one of claims 1 to 11 is implemented.