

MICROSERVICE CALL METHOD AND APPARATUS, DEVICE, AND MEDIUM

[0001] This application claims priority to Chinese Patent Application No. 202010168547.8, filed with the China National Intellectual Property Administration on March 12, 2020 and entitled
5 "MICROSERVICE CALL METHOD AND APPARATUS, SERVER, AND STORAGE MEDIUM", which is incorporated herein by reference in its entirety.

[0002] This application further claims priority to Chinese Patent Application No. 202010212312.4, filed with the China National Intellectual Property Administration on March 24, 2020 and entitled "MICROSERVICE CALL METHOD AND APPARATUS, DEVICE, AND
10 MEDIUM", which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0003] This application relates to the field of computer technologies, and in particular, to a microservice call method and apparatus, a device, and a computer-readable storage medium.

BACKGROUND

15 [0004] When developers use conventional monolithic architectures (monolithic architecture) to develop applications, for example, large-scale applications such as a customer relationship management (customer relationship management, CRM) system and an enterprise resource planning (enterprise resource planning, ERP) system, it becomes more difficult for the developers to update and repair these large-scale monolithic applications as new requirements continuously
20 increase. Based on this, the industry proposes a new architecture, that is, a microservice architecture (microservice architecture).

[0005] The microservice architecture aims to break down functions into discrete services to decouple solutions. In this way, developers can easily update and repair applications. In addition, a fault of one service does not affect an entire application, so that robustness of the application is
25 improved.

[0006] Currently, in the industry, a service mesh is mainly used (that is, an independent proxy, which is also referred to as a side car side car, is configured for each microservice) to migrate an application to a microservice framework for development. However, this method greatly increases a service call delay and reduces application performance.

5

SUMMARY

[0007] This application provides a microservice call method, to resolve a problem of application performance deterioration caused by a sharp increase in a service call delay in a conventional microservice framework. This application further provides an apparatus, a device, a computer-readable storage medium, and a computer program product corresponding to the microservice call method.

10

[0008] According to a first aspect, this application provides a microservice call method. In this method, functions to be implemented by distributed proxies are implemented by a network interface card. Specifically, the network interface card receives service call requests separately generated by a plurality of first services deployed on a first device, where each of the plurality of first services is corresponding to one second service, and the second service is specifically used to process first data of a service call request generated by the first service corresponding to the second service. Then, the network interface card sends the service call request to the second service based on service governance logic related to the first data. The network interface card is used to replace a proxy in a microservice framework, so that low intrusiveness is maintained. In addition, the network interface card can process together service call requests initiated by different services. This avoids a problem that contention of proxies for a system resource such as a central processing unit (central processor unit, CPU) causes process context switching and further causes an increase in a service call delay, and improves application performance.

15

20

[0009] In addition, processing the service call requests together by using the network interface card avoids occupation of a large quantity of resources caused by contention of different proxies, and improves resource utilization.

25

[0010] In some possible implementations, there may be a plurality of devices on which second services are deployed. Therefore, the network interface card may determine, based on the service governance logic related to the first data, such as load balancing logic, an address of a second

device providing the second service, and then the network interface card sends the service call request to the second service based on the address of the second device. In this way, functions such as service discovery, load balancing, encryption, identity authentication, authorization, and circuit breaker can be decoupled from a microservice, and low intrusiveness can be maintained.

5 **[0011]** In some possible implementations, in consideration of security, the network interface card may further perform traffic check for the service call request, to avoid a case in which a service provider cannot normally provide a service because an outlaw frequently sends service call requests. Specifically, the network interface card may send a traffic check request to a governance center, where the traffic check request is used to perform traffic check for N service call requests, and N is a positive integer. Then, the network interface card receives a traffic check result sent by the governance center, and sends the service call request to the second service when the traffic check result indicates that the check succeeds. The network interface card is used to send service call requests to the governance center in batches for traffic check. This can avoid extra transmission overheads caused when different proxies separately send service call requests to the governance center, and reduce overheads of interaction between the network interface card and the governance center in terms of data.

10 **[0012]** In some possible implementations, in consideration of traceability, the network interface card may further generate a service call log for query and use when a service is abnormal. Specifically, the network interface card may receive a service call response sent by the second service, where the service call response includes second data obtained by processing the first data by the second service. Then, the network interface card may generate a service call log based on at least one of the service call request and the service call response. Subsequently, the network interface card sends the service call log to a data collection center in batches. Compared with a method in which different proxies separately send service call logs to the data collection center, this method can greatly reduce overheads of interaction between the network interface card and the data collection center in terms of data.

25 **[0013]** In some possible implementations, in consideration that an exception such as breakdown may occur on a device (that is, a service provider) that provides a service, to avoid sending the service call request to an invalid service provider, the network interface card may further subscribe to a change message for a service provider list. In this way, the network interface card may determine, from a changed service provider list, the address of the second device

providing the second service. In this way, service reliability can be improved.

[0014] In some possible implementations, in consideration that a service governance rule may change, the network interface card may further subscribe to a change message for the service governance rule. In this way, the network interface card may run the service governance logic for the first data according to a changed service governance rule, to perform service governance.

[0015] In some possible implementations, the network interface card is specifically a smart network interface card. The smart network interface card is a programmable network interface card, and may be configured to assist a CPU in processing network traffic and the like. In an example, the smart network interface card may be specifically a network interface card in which a field-programmable gate array (field-programmable gate array, FPGA) is embedded.

[0016] According to a second aspect, this application provides a microservice call method. In this method, functions to be implemented by distributed proxies are implemented by a network interface card. Specifically, the network interface card receives service call requests sent by a plurality of first services, where each of the plurality of first services is corresponding to one second service, and the second service is used to process first data of a service call request generated by the first service corresponding to the second service. Then, the network interface card sends the service call request to the second service. Subsequently, the network interface card receives a service call response sent by the second service, where the service call response includes second data obtained by processing the first data by the second service. Finally, the network interface card sends the service call response to the first service corresponding to the second service.

[0017] The network interface card is used to replace a proxy in a microservice framework, so that low intrusiveness is maintained. In addition, the network interface card can process together service call requests initiated by different services. This avoids a problem that contention of proxies for a system resource such as a CPU causes process context switching and further causes an increase in a service call delay, and improves application performance.

[0018] In some possible implementations, the network interface card generates a service call log based on at least one of the service call request and the service call response, and then the network interface card sends the service call log to a data collection center in batches. Compared with a method in which different proxies separately send service call logs to the data collection center, this method can greatly reduce overheads of interaction between the network interface card

and the data collection center in terms of data.

[0019] In some possible implementations, the network interface card is specifically a smart network interface card. The smart network interface card is programmable. In an example, the smart network interface card may be a network interface card in which an FPGA is embedded.

5 **[0020]** According to a third aspect, this application provides a microservice call apparatus. The apparatus includes:

a communications module, configured to receive service call requests separately generated by a plurality of first services deployed on a first device, where each of the plurality of first services is corresponding to one second service, and the second service is used to process first
10 data of a service call request generated by the first service corresponding to the second service.

[0021] The communications module is further configured to send the service call request to the second service based on service governance logic related to the first data.

[0022] In some possible implementations, the communications module is specifically configured to:

15 determine, based on the service governance logic related to the first data, an address of a second device providing the second service; and

send the service call request to the second service based on the address of the second device.

[0023] In some possible implementations, the communications module is further configured
20 to:

send a traffic check request to a governance center, where the traffic check request is used to perform traffic check for N service call requests, and N is a positive integer;

receive a traffic check result sent by the governance center; and

25 send the service call request to the second service when the traffic check result indicates that the check succeeds.

[0024] In some possible implementations, the communications module is further configured to:

receive a service call response sent by the second service, where the service call response includes second data obtained by processing the first data by the second service.

30 **[0025]** The apparatus further includes:

a generation module, configured to generate a service call log based on at least one of

the service call request and the service call response.

[0026] The communications module is further configured to:

send the service call log to a data collection center in batches.

[0027] In some possible implementations, the apparatus further includes:

5 a subscription module, configured to subscribe to a change message for a service provider list, to determine, from a changed service provider list, the address of the second device providing the second service.

[0028] In some possible implementations, the apparatus further includes:

10 the subscription module, configured to subscribe to a change message for a service governance rule; and

a service governance module, configured to run the service governance logic for the first data according to a changed service governance rule.

[0029] In some possible implementations, the network interface card is a smart network interface card.

15 **[0030]** According to a fourth aspect, this application provides a microservice call apparatus. The apparatus includes:

a communications module, configured to receive service call requests sent by a plurality of first services, where each of the plurality of first services is corresponding to one second service, and the second service is used to process first data of a service call request
20 generated by the first service corresponding to the second service.

[0031] The communications module is further configured to send the service call request to the second service.

[0032] The communications module is further configured to receive a service call response sent by the second service, where the service call response includes second data obtained by
25 processing the first data by the second service.

[0033] The communications module is further configured to send the service call response to the first service corresponding to the second service.

[0034] In some possible implementations, the apparatus further includes:

30 a generation module, configured to generate a service call log based on at least one of the service call request and the service call response.

[0035] The communications module is further configured to send the service call log to a data

collection center in batches.

[0036] In some possible implementations, the network interface card is a smart network interface card.

[0037] According to a fifth aspect, this application provides a network interface card. The network interface card includes a processor and a memory. The processor and the memory communicate with each other. The processor is configured to execute instructions stored in the memory, so that the network interface card performs the microservice call method according to any one of the first aspect or the implementations of the first aspect.

[0038] According to a sixth aspect, this application provides a network interface card. The network interface card includes a processor and a memory. The processor and the memory communicate with each other. The processor is configured to execute instructions stored in the memory, so that the network interface card performs the microservice call method according to any one of the second aspect or the implementations of the second aspect.

[0039] According to a seventh aspect, this application provides a computer-readable storage medium. The computer-readable storage medium includes instructions, and the instructions instruct a network interface card to perform the microservice call method according to any one of the first aspect or the implementations of the first aspect.

[0040] According to an eighth aspect, this application provides a computer-readable storage medium. The computer-readable storage medium includes instructions, and the instructions instruct a network interface card to perform the microservice call method according to any one of the second aspect or the implementations of the second aspect.

[0041] According to a ninth aspect, this application provides a computer program product including instructions. When the computer program product runs on a network interface card, the network interface card is enabled to perform the microservice call method according to any one of the first aspect or the implementations of the first aspect.

[0042] According to a tenth aspect, this application provides a computer program product including instructions. When the computer program product runs on a network interface card, the network interface card is enabled to perform the microservice call method according to any one of the second aspect or the implementations of the second aspect.

[0043] In this application, based on the implementations according to the foregoing aspects, the implementations may be further combined to provide more implementations.

BRIEF DESCRIPTION OF DRAWINGS

[0044] To describe the technical methods in embodiments of this application more clearly, the following briefly describes the accompanying drawings for embodiments.

[0045] FIG. 1 is a diagram of a system architecture of a microservice call method according to an embodiment of this application;

[0046] FIG. 2 is an interaction flowchart of a microservice call method according to an embodiment of this application;

[0047] FIG. 3 is a diagram of processing logic of a microservice call method according to an embodiment of this application;

[0048] FIG. 4 is a schematic diagram of a structure of a microservice call apparatus according to an embodiment of this application;

[0049] FIG. 5 is a schematic diagram of a structure of a microservice call apparatus according to an embodiment of this application;

[0050] FIG. 6 is a schematic diagram of a structure of a device according to an embodiment of this application; and

[0051] FIG. 7 is a schematic diagram of a structure of a device according to an embodiment of this application.

DESCRIPTION OF EMBODIMENTS

[0052] The following describes the solutions in embodiments provided in this application with reference to the accompanying drawings in this application.

[0053] In the specification, claims, and accompanying drawings of this application, the terms "first", "second", and so on are intended to distinguish between similar objects but do not necessarily indicate a specific order or sequence. It should be understood that the terms used in such a way are interchangeable in appropriate circumstances. This is merely a discrimination manner that is used when objects having a same attribute are described in embodiments of this application.

[0054] To facilitate understanding of the technical solutions in this application, the following describes some technical terms in this application.

[0055] A microservice (microservice) is a small service including a single application. Each

microservice has its own process and can communicate with another microservice through a lightweight communication mechanism, such as an application programming interface (application programming interface, API) based on the hypertext transfer protocol (hypertext transfer protocol, HTTP).

5 **[0056]** A microservice architecture (microservice architecture) is an architecture that splits an application (especially a large-scale application, such as an e-commerce system) into several microservices. In some examples, the microservice architecture may specifically include a classic architecture such as Spring Cloud or Dubbo with low intrusiveness. Compared with a conventional monolithic architecture that integrates all functional services in one container, the microservice
10 architecture implements service decoupling by decomposing functions into discrete services. Service teams separately maintain microservices corresponding to respective services. This reduces maintenance complexity and improves maintenance efficiency. In addition, when a single microservice is faulty, another microservice can still work. This improves system stability.

[0057] In some cases, there may be a call relationship between a plurality of microservices of
15 an application. For example, in an e-commerce system, a microservice of a commodity system may further call a microservice of an evaluation system, to display a historical evaluation of a user on a commodity when commodity details are displayed. Service governance is to process a service call relationship in a distributed service framework. Service governance may be specifically divided into the following parts: service registration and discovery, load balancing, fault handling
20 and recovery (rate limiting, circuit breaker, timeout, and retry), gray release, service tracing, and the like.

[0058] Currently, an application based on the microservice architecture mainly implements service governance by using a service mesh (service mesh). The service mesh is an infrastructure layer used to process communication between services (including microservices). An
25 implementation of the service mesh is to allocate an independent proxy to each microservice. The proxy is also referred to as a side car (side car). The proxy is responsible for processing communication, monitoring, and some security-related work between services, to provide functions such as service discovery, load balancing, encryption, identity authentication, authorization, and circuit breaker. The microservice is no longer responsible for processing
30 specific logic of a service request, for example, performing load balancing based on a service request, but needs to complete only service processing.

[0059] For a large-scale application, one physical machine usually runs containers of a plurality of proxies, for example, containers of hundreds of proxies. However, a quantity of processes of the hundreds of proxies is far greater than a quantity of central processing unit (central processor unit, CPU) cores of the physical machine. When a CPU becomes a system bottleneck
5 due to heavy concurrent request pressure, frequent process context switching is inevitably caused because a plurality of processes contend for the CPU. As a result, a service call delay is sharply increased, and application performance is reduced.

[0060] Therefore, an embodiment of this application provides a microservice call method. In the method, a network interface card is used to replace a proxy of a microservice, and functions to
10 be implemented by distributed proxies are implemented together by the network interface card. Specifically, the network interface card may receive a service call request generated by at least one first service deployed on a device (which is referred to as a first device in this application). Each of a plurality of first services is corresponding to one second service. The second service is specifically used to process first data of a service call request generated by the first service
15 corresponding to the second service. Then, the network interface card may send the service call request to the second service based on service governance logic related to the first data, to implement microservice calling.

[0061] In the foregoing method, functions such as service discovery, load balancing, encryption, identity authentication, authorization, and circuit breaker are decoupled from the
20 microservice by using the network interface card, and low intrusiveness is maintained. In addition, the network interface card can process service call requests of different services together. This avoids a sharp increase in a service delay caused when proxies of different services contend for system resources including a CPU, and improves application performance.

[0062] In addition, functions implemented by proxies of different services are centralized on
25 the network interface card. This reduces application complexity and facilitates operation and maintenance, and management and control of the application. Further, performing centralized processing by using the network interface card can implement traffic check in batches. In this way, overheads of interaction with a governance center in terms of data are reduced, and concurrency pressure of the governance center in terms of management is reduced. Certainly, performing
30 centralized processing by using the network interface card can also report logs in batches. In this way, overheads of interaction with a data collection center in terms of data are reduced, and

concurrency pressure of the data collection in terms of management is reduced.

[0063] FIG. 1 shows a possible application scenario according to an embodiment of this application. In this application scenario, a device 104 and a device 124 are network interface cards. The device 104 is programmable, and may be programmed to implement customized processing logic, for example, implement processing logic for a service call request. Certainly, the device 124 may also be programmable. In actual application, the device 104 may be a smart network interface card (smart network interface card, smart NID), and the device 124 may be a common network interface card or a smart network interface card. The smart network interface card is a programmable network interface card, and the common network interface card is not a programmable network interface card.

[0064] A device 102 and a device 122 are computing devices. The computing device may be a physical machine, or may be a virtual machine obtained after virtualization. In actual application, the device 102 and the device 122 may be servers or cloud servers. A plurality of microservices are deployed on the device 102 and the device 122. Specifically, microservices deployed on the device 102 include a service 11, a service 12, ..., and a service 1n. Microservices deployed on the device 122 include a service 21, a service 22, ..., and a service 2n.

[0065] In this application scenario, there may further be a device 142, a device 162, and a device 182. Similarly, the device 142, the device 162, and the device 182 are computing devices, and the computing device may be a server or a cloud server. A registration center 1420 is deployed on the device 142. The registration center 1420 records a mapping relationship between a service and a service address. When a microservice calls another microservice, a service address of the called service may be obtained from the registration center 1420 for service calling. A governance center 1620 is deployed on the device 162, and the governance center 1620 records a service governance rule. When a microservice calls another microservice, the service governance rule may be obtained from the governance center 1620, and service governance logic is run based on the service governance rule. A data collection center 1820 is deployed on the device 182, and the data collection center 1820 records a service call log. When a microservice is faulty, an operation and maintenance engineer may perform troubleshooting and maintenance based on the service call log.

[0066] In this scenario, a microservice on the device 102 is a service caller, and a microservice on the device 122 is a service callee. For ease of differentiation, the device 102 may also be referred to as a first device, and the device 122 may also be referred to as a second device.

[0067] During specific implementation, a plurality of microservices (which are referred to as first services in this application) deployed on the device 102 may separately generate service call requests. Each first service is corresponding to one second service. The second service is specifically used to process first data of a service call request generated by the first service
5 corresponding to the second service. Then, the first services separately send the service call requests. A microservice call apparatus 1040 is deployed on the device 104 connected to the device 102. The microservice call apparatus 1040 is configured to receive a service call request generated by each of the plurality of first services such as the service 11, the service 12, ..., and the service 1n deployed on the device 102, and then send the service call request to the second service based
10 on service governance logic related to the first data.

[0068] A microservice call apparatus 1240 is deployed on the device 124. The microservice call apparatus 1240 receives the service call request sent by each of the plurality of first services, and then sends the service call request to the second service deployed on the device 122. In response to the service call request, the second service processes first data of the service call
15 request generated by the first service corresponding to the second service, to obtain second data. The second service may generate a service call response based on the second data, and send the service call response to the microservice call apparatus 1240. When receiving the service call response, the microservice call apparatus 1240 sends the service call response to the first service to implement microservice calling.

[0069] In actual application, the microservice call apparatus 1040 deployed on the device 104 may further send a service provider list change subscription request to the device 142. In this way, when a service provider changes, the registration center 1420 deployed on the device 142 may send a change message for a service provider list to the microservice call apparatus 1040 deployed
20 on the device 104. In this way, when determining, from the service provider list, an address of the second device providing the second service, the microservice call apparatus 1040 on the device 104 may determine, based on a changed service provider list, the address of the second device providing the second service. This avoids sending a service call request to an invalid service provider, and improves service reliability.

[0070] Similarly, the microservice call apparatus 1040 deployed on the device 104 may further
30 send a service governance rule change subscription request to the device 162. In this way, when the service governance rule changes, the governance center 1620 on the device 162 may send a

change message for the service governance rule to the microservice call apparatus 1040 deployed on the device 104. In this way, when running the service governance logic for the first data according to the service governance rule, the microservice call apparatus 1040 on the device 104 may run the service governance logic for the first data according to a changed service governance rule, to perform service governance.

[0071] It should be further noted that the foregoing scenario is described merely by using an example in which the microservice deployed on the device 102 is a service caller, and the microservice deployed on the device 122 is a service callee. In some implementations, the microservice deployed on the device 122 may alternatively be used as a service caller. When the microservice deployed on the device 122 is used as the service caller, the microservice call apparatus 1240 has a structure similar to that of the microservice call apparatus 1040. In addition, the microservice deployed on the device 102 may alternatively be used as a service callee. This is not limited in this embodiment of this application.

[0072] To facilitate understanding of the technical solutions of this application, the following describes a microservice call method in embodiments of this application from a perspective of interaction between the first service, the microservice call apparatus 1040, the microservice call apparatus 1240, the second service, the registration center 1420, the governance center 1620, and the data collection center 1820.

[0073] FIG. 2 is a flowchart of a microservice call method, and the method includes the following steps.

[0074] S202: The microservice call apparatus 1040 subscribes to a change message for a service provider list.

[0075] Specifically, the microservice call apparatus 1040 deployed on the device 104 may send a service provider list change subscription request to the registration center 1420 on the device 142. The service provider list change subscription request is used to request to subscribe to the change message for the service provider list. The service provider list change subscription request includes an identifier of the device 104. The registration center 1420 receives the service provider list change subscription request, and may add the identifier of the device 104 to a subscriber list. When the service provider list changes, the registration center 1420 may send the change message for the service provider list to a subscriber in the subscriber list. In this way, the microservice call apparatus 1040 deployed on the device 104 may obtain the change message for the service provider

list.

[0076] It should be understood that in some embodiments, S202 may not be performed when the microservice call method is performed. For example, the microservice call apparatus 1040 may pre-obtain the service provider list, and perform a subsequent operation based on the pre-obtained service provider list.

[0077] S204: The microservice call apparatus 1040 subscribes to a change message for a service governance rule.

[0078] Service governance refers to processing of a service call relationship in a distributed service framework, and specifically includes service registration and discovery, load balancing, fault handling and recovery (rate limiting, circuit breaker, timeout, and retry), gray release, service tracing, and the like. Based on this, the governance center 1620 may define a rule for service governance, that is, the service governance rule. The service governance rule may include a load balancing rule, a fault handling and recovery rule, and the like.

[0079] During specific implementation, the microservice call apparatus 1040 deployed on the device 104 may send a service governance rule change subscription request to the governance center 1620 on the device 162. The service governance rule change subscription request is specifically used to request the change message for the service governance rule. Similarly, the service governance rule change subscription request includes the identifier of the device 104. When receiving the service governance rule change subscription request, the governance center 1620 adds the identifier of the device 104 to a subscriber list. When the service governance rule changes, the governance center 1620 may send the change message for the service governance rule to a subscriber in the subscriber list. In this way, the microservice call apparatus 1040 deployed on the device 104 may obtain the change message for the service governance rule.

[0080] It should be noted that the subscriber list maintained by the governance center 1620 is a subscriber list for the service governance rule, and the subscriber list maintained by the registration center 1420 is a subscriber list for a service provider address. That is, the governance center 1620 and the registration center 1420 maintain different subscriber lists.

[0081] It should be understood that in some embodiments, S204 may not be performed when the microservice call method is performed. For example, the microservice call apparatus 1040 may pre-obtain the service governance rule, and perform a subsequent operation based on the pre-obtained service governance rule.

[0082] S206: A plurality of first services deployed on the device 102 separately send service call requests.

[0083] Each of the plurality of first services (for example, the service 11, the service 12, ..., and the service 1n shown in FIG. 1) is corresponding to one second service. The second service is specifically used to process first data of a service call request generated by the first service corresponding to the second service.

[0084] It should be noted that the first data is specifically data that needs to be processed by the second service. First data that needs to be processed by all services initiating service call requests may be the same or may be different. This may be specifically determined according to an actual requirement. Similarly, second services requested by the first services initiating the service call requests may be the same or may be different. This is not limited in this embodiment of this application.

[0085] S208: The microservice call apparatus 1040 determines, based on service governance logic related to the first data, an address of a second device providing the second service.

[0086] The service call request sent by the first service passes through the device 104. When receiving the service call request, the microservice call apparatus 1040 deployed on the device 104 may unload data of the service call request to obtain the first data. The first data includes at least a header of a data packet. In some cases, the first data may further include a body of the data packet. Then, the microservice call apparatus 1040 runs, based on the subscribed change message for the service governance rule in S204, the service governance logic for the first data by using a latest (that is, latest changed) service governance rule, to determine, from the service provider list, the address of the second device providing the second service.

[0087] It should be noted that when running the service governance logic for the first data to determine, from the service provider list, the address of the second device providing the second service, the microservice call apparatus 1040 may determine, from a latest (that is, latest changed) service provider list based on the subscribed change message for the service provider list in S202, the address of the second device providing the second service.

[0088] It should be understood that when S202 is not performed in the microservice call method, the microservice call apparatus 1040 may determine, based on the pre-obtained service provider list, the address of the second device providing the second service. When S204 is not performed in the microservice call method, the microservice call apparatus 1040 may run the

service governance logic for the first data according to the pre-obtained service governance rule.

[0089] That the microservice call apparatus 1040 runs the service governance logic for the first data according to the service governance rule is performing service governance on the first data according to the service governance rule. For ease of understanding, this application further provides a specific example for description. In an example, as shown in FIG. 3, in an application scenario of an e-commerce system, a microservice, that is, a service 11, in a commodity system initiates a service call request, and the service call request indicates to process data of a commodity A by using an evaluation obtaining service, that is, a service 21, in an evaluation system, to obtain a historical evaluation on the commodity A. Devices that provide the evaluation obtaining service include a device 1 in East China, a device 2 in North China, a device 3 in South China, and a device 4 in Central China. Traffic load of the device 1 and traffic load of the device 2 are higher than traffic load of the device 3 and traffic load of the device 4, and the traffic load of the device 4 is the lowest. In this case, the microservice call apparatus 1040 on a smart network interface card determines, according to a load balancing rule, that an address of the device 4 is an address of a device that finally provides the evaluation obtaining service, and sends the service call request to a service at the address.

[0090] In actual application, the microservice call apparatus 1040 may create a worker thread, and implement, by using the worker thread, the operation of running the service governance logic to determine, from the service provider list, the address of the second device providing the second service.

[0091] S210: The microservice call apparatus 1040 sends a traffic check request to the governance center 1620.

[0092] During specific implementation, the microservice call apparatus 1040 may add the service call request and the address of the corresponding second device to a traffic check queue. The traffic check queue is specifically a blocking queue. Then, the microservice call apparatus 1040 may send the traffic check request to the governance center 1620 in batches.

[0093] Specifically, the microservice call apparatus 1040 may create a checker thread, that is, a checker thread. The checker thread may extract, from the traffic check queue, all service call requests (a plurality of service call requests initiated by a plurality of services deployed on the device 102) waiting for traffic check, generate a traffic check request, and send the traffic check request to the governance center 1620. In this way, the service call requests are sent to the

governance center 1620 in batches.

[0094] S212: The microservice call apparatus 1040 receives a traffic check response returned by the governance center 1620. When a traffic check result in the traffic check response indicates that the check succeeds, S214 is performed.

5 **[0095]** The governance center 1620 may perform traffic check in batches for a plurality of service call requests in the traffic check request sent by the microservice call apparatus 1040, for example, may determine, by checking a source address or the like, whether the service call request is normal traffic or abnormal traffic generated by an attack performed by an outlaw. Then, the governance center 1620 may generate the traffic check response based on the traffic check result,
10 and return the traffic check response to the microservice call apparatus 1040.

[0096] It should be noted that, after the microservice call apparatus 1040 receives the traffic check response returned by the governance center 1620, the checker thread may further wake up a worker thread corresponding to this batch of service call requests, and then block a new service call request (that is, a service call request that has not been sent to the governance center 1620)
15 waiting in the traffic check queue.

[0097] When the traffic check result in the traffic check response received by the microservice call apparatus 1040 indicates that the check succeeds, the microservice call apparatus 1040 may perform a subsequent operation, for example, perform S214. When the traffic check result in the traffic check response received by the microservice call apparatus 1040 indicates that the check
20 fails, the microservice call apparatus 1040 may end the current microservice call procedure.

[0098] It should be understood that S210 and S212 are only optional implementations for improving security. In some embodiments, neither S210 nor S212 may be performed when the microservice call method is performed. In other words, the microservice call apparatus 1040 may perform S214 after performing S208.

25 **[0099]** S214: The microservice call apparatus 1040 sends the service call request to the second service based on the address of the second device.

[00100] After determining the address of the second device providing the second service, the microservice call apparatus 1040 may set a destination address in the service call request to the address of the second device providing the second service, that is, the device 122, and then send
30 the service call request to the second service.

[00101] S208 and S214 are an implementation in which the microservice call apparatus 1040

on the network interface card sends the service call request to the second service based on the service governance logic related to the first data. In another possible implementation of this embodiment of this application, the microservice call apparatus 1040 may send the service call request to the second service in another manner.

5 **[00102]** It should be noted that when the address, determined in S208, of the second device providing the second service is different from a destination address of the service call request in S206, the service call request sent by the microservice call apparatus 1040 in S214 is different from the service call request sent in S206. A main difference lies in that the destination address of the service call request sent in S214 is the address, re-determined in S208, of the second device
10 providing the second service. Certainly, in some cases, a packet body of the service call request sent in S214 may also be different from a packet body of the service call request sent in S206.

[00103] S216: The microservice call apparatus 1240 sends the service call request to the second service when receiving the service call request.

[00104] The service call request sent by the microservice call apparatus 1040 to the second
15 service passes through the device 124. When receiving the service call request, the microservice call apparatus 1240 deployed on the device 124 sends the service call request to the second service. It should be noted that, in consideration of network heterogeneity, when receiving the service call request, the microservice call apparatus 1240 may first unload the service call request, then regenerate a service call request based on unloaded data, and then send the regenerated service call
20 request to the second service.

[00105] S218: The second service returns a service call response.

[00106] When receiving the service call request, the second service may process the first data in response to the service call request, to obtain second data. For example, if the service call request indicates the evaluation obtaining service in the evaluation system to process the commodity A to
25 obtain the historical evaluation on the commodity A, the evaluation obtaining service may obtain the historical evaluation on the commodity A from all historical evaluations in response to the service call request.

[00107] The second service may generate the service call response according to the second data, where the service call response carries the second data, and then the second service returns the
30 service call response.

[00108] S220: The microservice call apparatus 1240 sends the service call response to the

microservice call apparatus 1040 when receiving the service call response.

[00109] The service call response returned by the second service passes through the device 124. When receiving the service call response, the microservice call apparatus 1240 deployed on the device 124 sends the service call response to the microservice call apparatus 1040. It should be noted that, similar to the service call request, when receiving the service call response, the microservice call apparatus 1240 may first unload the service call response, then regenerate a service call response based on unloaded data, and send the regenerated service call response to the microservice call apparatus 1040, so that the microservice call apparatus 1040 can identify the service call response.

[00110] S222: The microservice call apparatus 1040 sends the service call response to the first service corresponding to the second service.

[00111] Specifically, when receiving the service call response, the microservice call apparatus 1040 may unload the service call response, then regenerate a service call response based on unloaded data, and send the regenerated service call response to the first service.

[00112] In this way, the first service calls the second service. It should be noted that service governance is completed on the device 104, and there is no intrusion into service logic. For example, when gray release is performed on the second service, neither call logic of the first service nor implementation logic of the second service needs to be modified, and gray release processing logic only needs to be completed on the device 104.

[00113] S224: The microservice call apparatus 1040 sends a service call log to the data collection center 1820.

[00114] In some possible implementations, the microservice call apparatus 1040 deployed on the device 104 may further generate the service call log based on at least one of the received service call request and the received service call response, and then send the service call log to the data collection center 1820 in batches. The microservice call apparatus 1040 may further set a log uploading cycle. In this way, the microservice call apparatus 1040 may send the service call log to the data collection center 1820 in batches based on the preset log uploading cycle.

[00115] S226: The microservice call apparatus 1240 sends a service call log to the data collection center 1820.

[00116] In some possible implementations, the microservice call apparatus 1240 deployed on the device 124 may further generate the service call log based on at least one of the received service

call request and the received service call response, and then send the service call log to the data collection center 1820 in batches based on a preset cycle.

[00117] The microservice call apparatus 1040 and the microservice call apparatus 1240 upload the service call logs in batches, so that overheads of interaction between the device 104 and the data collection center 1820 in terms of data can be reduced, and overheads of interaction between the device 124 and the data collection center 1820 in terms of data can be reduced. This reduces costs.

[00118] The microservice call apparatus 1040 generates the service call log based on the received service call request and service call response, and the microservice call apparatus 1240 generates the service call log based on the received service call request and service call response, so that the data collection center 1820 can obtain complete log tracing data, including a time at which the first service initiates the service call request, a time at which the second service receives the service call request, a time at which the second service completes processing, and a time at which the first service receives the service call response. When some microservices are faulty, troubleshooting and maintenance can be performed based on the complete log tracing data.

[00119] It should be understood that S224 and S226 are merely optional implementations provided in this embodiment of this application. In some embodiments, neither S224 nor S226 may be performed when the microservice call method is performed.

[00120] In FIG. 2, steps identified by using dashed lines or dashed-line boxes are optional steps provided in this embodiment of this application. In some possible implementations, these steps may not be performed, or may be implemented in another manner. This is not limited in this embodiment of this application.

[00121] Based on the foregoing content description, in the microservice call method provided in this embodiment of this application, a network interface card is used to replace a proxy in a service mesh, and the network interface card processes together service call requests initiated by different services deployed on one device, to complete service governance. This avoids intrusion into service logic. In addition, the network interface card may process together service call requests initiated by different services. This avoids a problem of a sharp increase in a delay caused by contention of different services for a system resource including a CPU, reduces resource usage of an application, improves application response efficiency, and further improves application performance.

[00122] The foregoing describes in detail the microservice call method provided in embodiments of this application with reference to FIG. 1 to FIG. 3. The following describes, with reference to the accompanying drawings, a microservice call apparatus and a corresponding device provided in embodiments of this application.

5 **[00123]** FIG. 4 is a schematic diagram of a structure of a microservice call apparatus, and the microservice call apparatus 1040 includes:

a communications module 1042, configured to receive service call requests separately generated by a plurality of first services deployed on a first device, where each of the plurality of first services is corresponding to one second service, and the second service is used to process first
10 data of a service call request generated by the first service corresponding to the second service.

[00124] The communications module 1042 is further configured to send the service call request to the second service based on service governance logic related to the first data.

[00125] In some possible implementations, the communications module 1042 is specifically configured to:

15 determine, based on the service governance logic related to the first data, an address of a second device providing the second service; and

send the service call request to the second service based on the address of the second device.

[00126] In some possible implementations, the communications module 1042 is further
20 configured to:

send a traffic check request to a governance center, where the traffic check request is used to perform traffic check for N service call requests, and N is a positive integer;

receive a traffic check result sent by the governance center; and

25 send the service call request to the second service when the traffic check result indicates that the check succeeds.

[00127] In some possible implementations, the communications module is further configured to:

receive a service call response sent by the second service, where the service call response includes second data obtained by processing the first data by the second service.

30 **[00128]** The apparatus 1040 further includes:

a generation module 1044, configured to generate a service call log based on at least

one of the service call request and the service call response.

[00129] The communications module 1042 is further configured to:

send the service call log to a data collection center in batches.

[00130] In some possible implementations, the apparatus 1040 further includes:

5 a subscription module 1046, configured to subscribe to a change message for a service provider list, to determine, from a changed service provider list, the address of the second device providing the second service.

[00131] In some possible implementations, the apparatus 1040 further includes:

10 the subscription module 1046, configured to subscribe to a change message for a service governance rule; and

a service governance module 1048, configured to run the service governance logic for the first data according to a changed service governance rule.

[00132] In some possible implementations, the network interface card is a smart network interface card.

15 **[00133]** The microservice call apparatus 1040 according to this embodiment of this application may correspondingly perform the methods described in embodiments of this application, and the foregoing and other operations and/or functions of the modules of the microservice call apparatus 1040 are separately used to implement corresponding procedures of the methods in the embodiment shown in FIG. 2. For brevity, details are not described herein again.

20 **[00134]** FIG. 5 is a schematic diagram of a structure of a microservice call apparatus, and the microservice call apparatus 1240 includes:

a communications module 1242, configured to receive service call requests sent by a plurality of first services, where each of the plurality of first services is corresponding to one second service, and the second service is used to process first data of a service call request
25 generated by the first service corresponding to the second service.

[00135] The communications module 1242 is further configured to send the service call request to the second service.

[00136] The communications module 1242 is further configured to receive a service call response sent by the second service, where the service call response includes second data obtained
30 by processing the first data by the second service.

[00137] The communications module 1242 is further configured to send the service call

response to the first service corresponding to the second service.

[00138] In some possible implementations, the apparatus 1240 further includes:

a generation module 1244, configured to generate a service call log based on at least one of the service call request and the service call response.

5 **[00139]** The communications module 1242 is further configured to send the service call log to a data collection center in batches.

[00140] In some possible implementations, the network interface card is a smart network interface card.

[00141] The microservice call apparatus 1240 according to this embodiment of this application
10 may correspondingly perform the methods described in embodiments of this application, and the foregoing and other operations and/or functions of the modules of the microservice call apparatus 1240 are separately used to implement corresponding procedures of the methods in the embodiment shown in FIG. 2. For brevity, details are not described herein again.

[00142] An embodiment of this application further provides a device. The device may be a
15 network interface card, and is specifically configured to implement a function of the microservice call apparatus 1040 in the embodiment shown in FIG. 4.

[00143] FIG. 6 is a schematic diagram of a structure of a device. As shown in FIG. 6, a device 600 includes a bus 601, a processor 602, a communications interface 603, and a memory 604. The processor 602, the memory 604, and the communications interface 603 communicate with each
20 other through the bus 601. The bus 601 may be a peripheral component interconnect (peripheral component interconnect, PCI) bus, an extended industry standard architecture (extended industry standard architecture, EISA) bus, or the like. Buses may be classified into an address bus, a data bus, a control bus, and the like. For ease of representation, only one bold line is used to represent the bus in FIG. 6, but this does not mean that there is only one bus or only one type of bus. The
25 communications interface 603 is configured to communicate with the outside. For example, the communications interface 603 is configured to receive a service call request, send a service call request, or the like.

[00144] The processor 602 may be a central processing unit (central processing unit, CPU). The memory 604 may include a volatile memory (volatile memory), for example, a random access memory (random access memory, RAM). Alternatively, the memory 604 may include a non-
30 volatile memory (non-volatile memory), for example, a read-only memory (read-only memory,

ROM), a flash memory, an HDD, or an SSD.

[00145] The memory 604 stores executable code, and the processor 602 executes the executable code to perform the foregoing microservice call method.

[00146] Specifically, when the embodiment shown in FIG. 4 is implemented, and the modules of the microservice call apparatus 1040 described in the embodiment of FIG. 4 are implemented by using software, software or program code required for performing functions of the generation module 1044, the subscription module 1046, and the service governance module 1048 in FIG. 4 is stored in the memory 604. A function of the communications module 1042 is implemented by using the communications interface 603. The communications interface 603 receives service call requests separately generated by a plurality of first services deployed on a first device, and transmits the service call requests to the processor 602 through the bus 601. The processor 602 executes program code that is corresponding to each module and that is stored in the memory 604, to perform the microservice call method, to send the service call requests to second services by using the communications interface 603 based on service governance logic related to first data, to implement microservice calling.

[00147] An embodiment of this application further provides another device. The device may be a network interface card, including a common network interface card or a smart network interface card, and is specifically configured to implement a function of the microservice call apparatus 1240 in the embodiment shown in FIG. 5.

[00148] FIG. 7 is a schematic diagram of a structure of a device. As shown in FIG. 7, a device 700 includes a bus 701, a processor 702, a communications interface 703, and a memory 704. The processor 702, the memory 704, and the communications interface 703 communicate with each other through the bus 701.

[00149] The memory 704 stores executable code, and the processor 702 executes the executable code to perform the foregoing microservice call method.

[00150] Specifically, when the embodiment shown in FIG. 5 is implemented, and the modules of the microservice call apparatus 1240 described in the embodiment of FIG. 5 are implemented by using software, software or program code required for performing a function of the generation module 1244 in FIG. 5 is stored in the memory 704. A function of the communications module 1242 is implemented by using the communications interface 703. The communications interface 703 receives service call requests separately sent by a plurality of first services, and transmits the

service call requests to the processor 702 through the bus 701. The processor 702 executes program code that is corresponding to each module and that is stored in the memory 704, to perform the microservice call method.

5 **[00151]** An embodiment of this application further provides a computer-readable storage medium. The computer-readable storage medium includes instructions, and the instructions instruct a network interface card to perform the foregoing microservice call method applied to the microservice call apparatus 1040.

10 **[00152]** An embodiment of this application further provides a computer-readable storage medium. The computer-readable storage medium includes instructions, and the instructions instruct a network interface card to perform the foregoing microservice call method applied to the microservice call apparatus 1240.

15 **[00153]** An embodiment of this application further provides a computer program product. When the computer program product is executed by a network interface card, the network interface card performs any one of the foregoing microservice call methods. The computer program product may be a software installation package. If any one of the foregoing microservice call methods needs to be used, the computer program product may be downloaded, and the computer program product may be executed on a network interface card.

CLAIMS

What is claimed is:

1. A microservice call method, wherein the method comprises:

receiving, by a network interface card, service call requests separately generated by a plurality
5 of first services deployed on a first device, wherein each of the plurality of first services is
corresponding to one second service, and the second service is used to process first data of a service
call request generated by the first service corresponding to the second service; and

sending, by the network interface card, the service call request to the second service based on
service governance logic related to the first data.

10 2. The method according to claim 1, wherein the sending, by the network interface card, the
service call request to the second service based on service governance logic related to the first data
comprises:

determining, by the network interface card based on the service governance logic related to
the first data, an address of a second device providing the second service; and

15 sending, by the network interface card, the service call request to the second service based on
the address of the second device.

3. The method according to claim 1, wherein the method further comprises:

sending, by the network interface card, a traffic check request to a governance center, wherein
the traffic check request is used to perform traffic check for N service call requests, and N is a
20 positive integer;

receiving, by the network interface card, a traffic check result sent by the governance center;
and

sending the service call request to the second service when the traffic check result indicates
that the check succeeds.

25 4. The method according to claim 1, wherein the method further comprises:

receiving, by the network interface card, a service call response sent by the second service,
wherein the service call response comprises second data obtained by processing the first data by
the second service;

generating, by the network interface card, a service call log based on at least one of the service
30 call request and the service call response; and

sending, by the network interface card, the service call log to a data collection center in batches.

5. The method according to any one of claims 1 to 4, wherein the method further comprises:

subscribing, by the network interface card, to a change message for a service provider list, to
5 determine, from a changed service provider list, the address of the second device providing the second service.

6. The method according to any one of claims 1 to 5, wherein the method further comprises:

subscribing, by the network interface card, to a change message for a service governance rule;
and

10 running, by the network interface card, the service governance logic for the first data according to a changed service governance rule.

7. The method according to any one of claims 1 to 6, wherein the network interface card is a smart network interface card.

8. A microservice call method, wherein the method comprises:

15 receiving, by a network interface card, service call requests sent by a plurality of first services, wherein each of the plurality of first services is corresponding to one second service, and the second service is used to process first data of a service call request generated by the first service corresponding to the second service;

sending, by the network interface card, the service call request to the second service;

20 receiving, by the network interface card, a service call response sent by the second service, wherein the service call response comprises second data obtained by processing the first data by the second service; and

sending, by the network interface card, the service call response to the first service corresponding to the second service.

25 9. The method according to claim 8, wherein the method further comprises:

generating, by the network interface card, a service call log based on at least one of the service call request and the service call response; and

sending, by the network interface card, the service call log to a data collection center in batches.

30 10. The method according to claim 8 or 9, wherein the network interface card is a smart network interface card.

11. A microservice call apparatus, wherein the apparatus comprises:

a communications module, configured to receive service call requests separately generated by a plurality of first services deployed on a first device, wherein each of the plurality of first services is corresponding to one second service, and the second service is used to process first data of a service call request generated by the first service corresponding to the second service; and

the communications module is further configured to send the service call request to the second service based on service governance logic related to the first data.

12. The apparatus according to claim 11, wherein the communications module is specifically configured to:

determine, based on the service governance logic related to the first data, an address of a second device providing the second service; and

send the service call request to the second service based on the address of the second device.

13. The apparatus according to claim 11, wherein the communications module is further configured to:

send a traffic check request to a governance center, wherein the traffic check request is used to perform traffic check for N service call requests, and N is a positive integer;

receive a traffic check result sent by the governance center; and

send the service call request to the second service when the traffic check result indicates that the check succeeds.

14. The apparatus according to claim 11, wherein the communications module is further configured to:

receive a service call response sent by the second service, wherein the service call response comprises second data obtained by processing the first data by the second service;

the apparatus further comprises:

a generation module, configured to generate a service call log based on at least one of the service call request and the service call response; and

the communications module is further configured to:

send the service call log to a data collection center in batches.

15. The apparatus according to any one of claims 11 to 14, wherein the apparatus further comprises:

a subscription module, configured to subscribe to a change message for a service provider

list, to determine, from a changed service provider list, the address of the second device providing the second service.

16. The apparatus according to any one of claims 11 to 15, wherein the apparatus further comprises:

5 the subscription module, configured to subscribe to a change message for a service governance rule; and

a service governance module, configured to run the service governance logic for the first data according to a changed service governance rule.

17. The apparatus according to any one of claims 11 to 16, wherein the microservice call
10 apparatus is a smart network interface card.

18. A microservice call apparatus, wherein the apparatus comprises:

a communications module, configured to receive service call requests sent by a plurality of first services, wherein each of the plurality of first services is corresponding to one second service, and the second service is used to process first data of a service call request generated by the first
15 service corresponding to the second service;

the communications module is further configured to send the service call request to the second service;

the communications module is further configured to receive a service call response sent by the second service, wherein the service call response comprises second data obtained by processing
20 the first data by the second service; and

the communications module is further configured to send the service call response to the first service corresponding to the second service.

19. The apparatus according to claim 18, wherein the apparatus further comprises:

a generation module, configured to generate a service call log based on at least one of the
25 service call request and the service call response; and

the communications module is further configured to send the service call log to a data collection center in batches.

20. The apparatus according to claim 18 or 19, wherein the network interface card is a smart network interface card.

30 21. A network interface card, wherein the network interface card comprises a processor and a memory, wherein

the processor is configured to execute instructions stored in the memory, to enable the network interface card to perform the microservice call method according to any one of claims 1 to 7.

22. A network interface card, wherein the network interface card comprises a processor and
5 a memory, wherein

the processor is configured to execute instructions stored in the memory, to enable the network interface card to perform the microservice call method according to any one of claims 8 to 10.

23. A computer-readable storage medium, wherein the computer-readable storage medium
10 comprises instructions, and the instructions instruct a network interface card to perform the microservice call method according to any one of claims 1 to 7.

24. A computer-readable storage medium, wherein the computer-readable storage medium comprises instructions, and the instructions instruct a network interface card to perform the microservice call method according to any one of claims 8 to 10.

15