CLAIMS

What is claimed is:

1. A node for running a container group, wherein a connection control module, a sidecar cluster, and a first service container group are run on the node, and the sidecar cluster comprises at least two sidecars, wherein

the connection control module is configured to: receive a sidecar allocation policy sent by a console connected to the node, select a first sidecar from the sidecar cluster according to the sidecar allocation policy, and forward, to the first sidecar, a data packet sent by the first service container group; and

the first sidecar is configured to perform traffic management on the data packet sent by the first service container group.

2. The node according to claim 1, further comprising a second service container group, wherein

the connection control module is further configured to: select a second sidecar from the sidecar cluster according to the sidecar allocation policy, and forward, to the second sidecar, a data packet sent by the second service container group; and

the second sidecar is configured to perform traffic management on the data packet sent by the second service container group.

3. The node according to claim 2, wherein a specification of a hardware resource allocated to the first sidecar is higher than that of a hardware resource allocated to the second sidecar, the sidecar allocation policy comprises a first policy, and the first policy indicates the first service container group to preferentially use the first sidecar; and

the connection control module is specifically configured to select the first sidecar from the sidecar cluster according to the first policy.

4. The node according to any one of claims 1 to 3, further comprising the second service container group, wherein the sidecar allocation policy further comprises a second policy, and the second policy indicates that a quantity of objects served by the first sidecar does not exceed an upper limit value;

the connection control module is further configured to: determine the quantity of objects served by the first sidecar, and forward, to the first sidecar when the quantity of objects served by the first sidecar does not exceed the upper limit value, the data packet sent by the second service container group; and

the first sidecar is further configured to simultaneously perform traffic management on the data packet sent by the first service container group and the data packet sent by the second service

container group.

5. The node according to any one of claims 1 to 4, wherein

the connection control module is configured to: after the first sidecar is faulty, select a third sidecar from the sidecar cluster, or notify the console to create the third sidecar in the node, and forward, to the third sidecar, another data packet sent by the first service container group; and

the third sidecar is configured to perform traffic management on the another data packet sent by the first service container group.

6. The node according to claim 5, wherein the third sidecar is a new version to which a function is upgraded based on the first sidecar, or the third sidecar is a replication version of the first sidecar.

7. The node according to claim 5 or 6, wherein

the first sidecar is configured to send the data packet to a back-end container group after performing traffic management on the data packet sent by the first service container group.

8. The node according to claim 7, wherein

the first sidecar is further configured to: generate a session identifier, and send the session identifier to the first service container group and the connection control module;

the connection control module is configured to record a correspondence between the session identifier and the back-end container group; and

the third sidecar is configured to: obtain the session identifier from the first service container group, determine, based on the session identifier, the back-end container group in the correspondence recorded by the connection control module, and send the another data packet to the back-end container group after performing traffic management on the another data packet sent by the first service container group.

9. The node according to claim 1 or 2, wherein the sidecar allocation policy comprises a third policy, and the third policy indicates that a sidecar in the sidecar cluster is preferentially used when a quantity of objects served by the sidecar is 0; and

the connection control module is further configured to: determine a quantity of objects served by the first sidecar, and forward, to the first sidecar when the quantity of objects served by the first sidecar is 0, the data packet sent by the first service container group.

10. A method for managing a container group in a node, wherein a connection control module, a sidecar cluster, and a first service container group are run on the node, the sidecar cluster comprises at least two sidecars, and the method comprises:

receiving, by the connection control module, a sidecar allocation policy sent by a console connected to the node, selecting a first sidecar from the sidecar cluster according to the sidecar

allocation policy, and forwarding, to the first sidecar, a data packet sent by the first service container group; and

performing, by the first sidecar, traffic management on the data packet sent by the first service container group.

11. The method according to claim 10, wherein a second service container group is further run on the node, and the method further comprises:

selecting, by the connection control module, a second sidecar from the sidecar cluster according to the sidecar allocation policy, and forwarding, to the second sidecar, a data packet sent by the second service container group; and

performing, by the second sidecar, traffic management on the data packet sent by the second service container group.

12. The method according to claim 11, wherein a specification of a hardware resource allocated to the first sidecar is higher than that of a hardware resource allocated to the second sidecar, the sidecar allocation policy comprises a first policy, and the first policy indicates the first service container group to preferentially use the first sidecar; and

the selecting a first sidecar from the sidecar cluster according to the sidecar allocation policy comprises: selecting the first sidecar from the sidecar cluster according to the first policy.

13. The method according to any one of claims 10 to 12, wherein the second service container group is further run on the node, the sidecar allocation policy further comprises a second policy, and the second policy indicates that a quantity of objects served by the first sidecar does not exceed an upper limit value; and the method further comprises:

determining, by the connection control module, the quantity of objects served by the first sidecar, and forwarding, to the first sidecar when the quantity does not exceed the upper limit value, the data packet sent by the second service container group; and

simultaneously performing, by the first sidecar, traffic management on the data packet sent by the first service container group and the data packet sent by the second service container group.

14. The method according to any one of claims 10 to 13, wherein the method further comprises:

after the first sidecar is faulty, selecting, by the connection control module, a third sidecar from the sidecar cluster, or notifying the console to create the third sidecar in the node, and forwarding, to the third sidecar, another data packet sent by the first service container group; and

performing, by the third sidecar, traffic management on the another data packet sent by the first service container group.

15. The method according to claim 18, wherein the third sidecar is a new version to which a function is upgraded based on the first sidecar, or the third sidecar is a replication version of the

first sidecar.

16. The method according to claim 14 or 15, wherein the method further comprises:

sending, by the first sidecar, the data packet to a back-end container group after performing traffic management on the data packet sent by the first service container group.

17. The method according to claim 16, wherein the method further comprises:

generating, by the first sidecar, a session identifier, and sending the session identifier to the first service container group and the connection control module;

recording, by the connection control module, a correspondence between the session identifier and the back-end container group; and

obtaining, by the third sidecar, the session identifier from the first service container group, determining, based on the session identifier, the back-end container group in the correspondence recorded by the connection control module, and sending the another data packet to the back-end container group after performing traffic management on the another data packet sent by the first service container group.

18. The method according to claim 10 or 11, wherein the sidecar allocation policy comprises a third policy, and the third policy indicates that a sidecar in the sidecar cluster is preferentially used when a quantity of objects served by the sidecar is 0; and

the selecting a first sidecar from the sidecar cluster according to the sidecar allocation policy, and forwarding, to the first sidecar, a data packet sent by the first service container group comprises:

determining, by the connection control module, a quantity of objects served by the first sidecar, and forwarding, to the first sidecar when the quantity of objects served by the first sidecar is 0, the data packet sent by the first service container group.

19. A node for running a container group, comprising a processor and a memory, wherein the processor is configured to execute instructions stored in the memory, to perform the method according to any one of claims 10 to 18.

20. A computer program product comprising instructions, wherein when the instructions are run by a computing device cluster, the computing device cluster is enabled to perform the method according to any one of claims 10 to 18.