CLAIMS

What is claimed is:

1. A method, comprising:

configuring a container-based serverless computing system such that an application instance is allowed to be in one of an online state and a low-power state at runtime, wherein the application instance consumes less power or resources in the low-power state than it does in the online state;

in response to scaling down an application, causing at least one first application instance of the application in the online state to enter the low-power state; the causing the at least one first application instance to enter the low-power state comprises causing a memory space occupied by the at least one first application instance to be within a memory range for low-power instances in a memory; and

in response to scaling up the application, causing at least one second application instance of the application in the low-power state to enter the online state, the causing the at least one second application instance to enter the online state comprises causing the memory space occupied by the at least one second application instance to be outside the memory range for the low-power instances in the memory.

2. The method according to claim 1, wherein the causing the at least one first application instance of the application in the online state to enter the low-power state comprises:

imposing a limit on a resource quota for the at least one first application instance; and

downgrading a resource configuration of a respective container where the at least one first application instance is located.

3. The method according to claim 2, further comprising:

in response to determining that the resource configuration of the respective container is downgraded, based on an in-place pod resource update mechanism, returning resources released by the respective container to a scheduling system.

4. The method according to claim 3, wherein a memory range for a low-power instance is a range outside an execution range of a periodic memory management operation in the memory.

5. The method according to claim 4, wherein the periodic memory management operation comprises a memory garbage collection operation or a memory release operation for releasing the memory that has not been used within a predetermined period.

6. The method according to claim 4, further comprising:

in response to a range adjustment instruction, adjusting the execution range or a size of the execution range; and

in response to determining that a first application instance occupying a memory space enters the low-power state, setting the execution range to exclude the memory space occupied by the first application instance.

7. The method according to claim 1, wherein the causing the at least one first application instance to enter the low-power state comprises:

based on a CPU sharing function, adding a plurality of the first application instances from the at least one first application instance to run in a CPU group with a low-power specification, wherein application instances in the CPU group with the low-power specification share one or more CPUs.

8. The method according to claim 1, wherein the causing the at least one second application instance of the application in the low-power state to enter the online state comprises:

upgrading a resource configuration of a respective container where the at least one second application instance is located; and

releasing a limit on a resource quota for the at least one second application instance.

9. The method according to claim 8, wherein the causing the at least one second application instance of the application in the low-power state to enter the online state further comprises:

causing, in priority, a second application instance in a container on a machine with idle resources in a node where the container is located to enter the online state.

10. The method according to claim 8, further comprising:

in response to determining that the application instance is to enter the online state and resources in a node where it is located are insufficient, migrating the application instance to a relatively idle node compared with a current node based on a live migration function of

a checkpoint/restore in userspace (CRIU); or

in response to determining that multiple application instances in the low-power state are located on a same node, making one or more of the multiple application instances into a checkpoint snapshot based on the CRIU and, based on the checkpoint snapshot, restoring the one or more of the multiple application instances on the relatively idle node.

11. The method according to claim 8, wherein

the limit on the resource quota for the application instance is imposed or released based on a control groups (Cgroups) mechanism; and

in response to determining that the resource configuration of the respective container is upgraded, based on an in-place pod resource update mechanism, requesting resources for the respective container from a scheduling system.

12. The method according to claim 1, wherein the causing the at least one second application instance to enter the online state comprises:

causing the at least one second application instance to exit a CPU group with a low-power specification.

13. The method according to claim 1, further comprising:

causing memory data of one or more application instances in the low-power state to be transferred between a memory and a storage device.

14. The method according to claim 13, wherein the causing the memory data of the one or more application instances in the low-power state to be transferred between the memory and the storage device comprises:

using a memory swap function in a kernel mode to cause the memory data of the one or more application instances in the low-power state to be transferred between the memory and the storage device; or

using a memory swap function in a user mode to cause the memory data of the one or more application instances in the low-power state to be transferred between the memory and the storage device and to cause memory data from different containers to be transferred to different storage devices or different pages of the storage device.

15. The method according to claim 14, wherein:

the memory data of multiple application instances in the low-power state are

simultaneously transferred from the memory to the different storage devices or the different pages of the storage device by using the memory swap function in the user mode; or

the memory data of multiple application instances in the low-power state are simultaneously transferred from the different storage devices or the different pages of the storage device to the memory.

16. The method according to claim 14, further comprising:

receiving a memory swap setting instruction for an application or an application instance, wherein the memory swap setting instruction is configured to indicate whether the memory swap function in the kernel mode or the memory swap function in user mode is used to perform memory data transfer for an application instance of the application or the application instance; and

in response to the memory swap setting instruction, setting the memory swap function used to perform the memory data transfer for the application instance of the application or the application instance.

17. The method according to claim 14, further comprising:

in response to determining that the memory swap function in user mode is used, using multiple enhanced SSDs to construct storage devices with a same priority for swap.

18. The method according to claim 13, wherein the causing the memory data of the one or more application instances in the low-power state to be transferred between the memory and the storage device comprises:

based on a least recently used algorithm, selecting the memory data of the one or more application instances in the low-power state, and transferring and persisting it from the memory into the storage device; and/or

in response to a swap-back instruction or a traffic request or an instance deployment strategy change, transferring the memory data of the one or more application instances in the low-power state on the storage device back to the memory.

19. The method according to claim 13, wherein after memory data of a second application instance has been transferred to the storage device, the causing the at least one second application instance of the application in the low-power state to enter the online state further comprises:

transferring the memory data of the second application instance from the storage

device to an execution range of a periodic memory management operation in the memory.

20. A computer device comprising:

one or more processors; and

one or more memories storing thereon computer-readable instructions that, when executed by the one or more processors, cause the one or more processors to perform acts comprising:

configuring a container-based serverless computing system such that an application instance is allowed to be in one of an online state and a low-power state at runtime, the application instance consuming less power or resources in the low-power state than it does in the online state;

in response to scaling down an application, causing at least one first application instance of the application in the online state to enter the low-power state, the causing the at least one first application instance to enter the low-power state comprising based on a CPU sharing function, adding a plurality of first application instances from the at least one first application instance to run in a CPU group with a low-power specification, application instances in the CPU group with the low-power specification sharing one or more CPUs; and

in response to scaling up the application, causing at least one second application instance of the application in the low-power state to enter the online state.

21. One or more memories storing thereon non-transitory computer-readable instructions that, when executed by one or more processors, cause the one or more processors to perform acts comprising:

configuring a container-based serverless computing system such that an application instance is allowed to be in one of an online state and a low-power state at runtime, the application instance consuming less power or resources in the low-power state than it does in the online state;

in response to scaling down an application, causing at least one first application instance of the application in the online state to enter the low-power state, the causing the at least one first application instance of the application in the online state to enter the low-power state comprising:

imposing a limit on a resource quota for the at least one first application instance;

downgrading a resource configuration of a respective container where the at least one first application instance is located; and

in response to determining that the resource configuration of the respective container is downgraded, based on an in-place pod resource update mechanism, returning resources released by the respective container to a scheduling system; and

in response to scaling up the application, causing at least one second application instance of the application in the low-power state to enter the online state,

wherein a memory range for a low-power instance is a range outside an execution range of a periodic memory management operation in a memory.