CLAIMS

1. A computer-implemented method, comprising:

defining at least one workload container for a respective node of a plurality of nodes, wherein the at least one workload container includes a cache component of a cache-mesh, wherein the cache component includes a local memory for the at least one workload container, wherein the cache-mesh includes a plurality of cache components across different nodes of the plurality of nodes, wherein the plurality of cache components includes a protocol for communication between each other;

in response to receiving a state request from a stateless requesting service from one of the plurality of cache components of the cache-mesh in an execution container, determining whether a requested state associated with the state request is present in the cache component of a related execution container;

in response to a cache miss, broadcasting the state request to other cache components of the plurality of cache components of the cache-mesh;

determining, by the other cache components of the plurality of cache components of the cache-mesh, whether the requested state is present in respective caches of the other cache components; and

in response to any cache component identifying the requested state, sending the requested state to the requesting service using the protocol for communication.

2. The method of claim 1, wherein the protocol for communication is encrypted.

3. The method of claim 1 or 2, further comprising:

in response to completion of an execution of the execution container on the respective node, broadcasting an end state to the cache-mesh.

4. The method of one of claims 1 to 3, further comprising:

providing a first centralized cache component for the plurality of nodes.

5. The method of claim 4, further comprising:

in response to determining that the requested state is not available in the plurality of cache components of the cache-mesh, loading the requested state from the first centralized cache component.

6. The method of one of claims 1 to 5, wherein the protocol for communication includes a component selected from the group consisting of an identifier of a user, a key identifying a cached value, and the cached value.

7. The method of claim 4, further comprising:

in response to completion of the execution of the execution container on the respective node, broadcasting the end state to the first centralized cache component.

8. The method of one of claims 1 to 7, wherein the plurality of cache components of the cache-mesh include a pool of cache components having a comparable workload type.

9. The method of claim 8, wherein the comparable workload type includes an execution pattern selected from the group consisting of an identical runtime engine, a user information, and an identical programming language.

10. The method of claim 4, further comprising:

synchronizing a second centralized cache component with the first centralized cache component using a replication protocol, wherein the second centralized cache component is maintained in a second remote workload execution environment.

11. A computer system for execution of a stateless service on a node in a workload execution environment, comprising:

one or more processors, one or more computer-readable memories, one or more computer-readable tangible storage media, and program instructions stored on at least one of the one or more computer-readable tangible storage media for execution by at least one of the one or more processors via at least one of the one or more memories, wherein the computer system is capable of performing a method comprising:

defining at least one workload container for a respective node of a plurality of nodes, wherein the at least one workload container includes a cache component of a cache-mesh,

wherein the cache component includes a local memory for the at least one workload container, wherein the cache-mesh includes a plurality of cache components across different nodes of the plurality of nodes, wherein the plurality of cache components includes a protocol for communication between each other;

in response to receiving a state request from a stateless requesting service from one of the plurality of cache components of the cache-mesh in an execution container, determining whether a requested state associated with the state request is present in the cache component of a related execution container;

in response to a cache miss, broadcasting the state request to other cache components of the plurality of cache components of the cache-mesh;

determining, by the other cache components of the plurality of cache components of the cache-mesh, whether the requested state is present in respective caches of the other cache components; and

in response to any cache component identifying the requested state, sending the requested state to the requesting service using the protocol for communication.

12. The system of claim 11, further comprising:

in response to completion of an execution of the execution container on the respective node, broadcasting an end state to the cache-mesh.

13. The system of claim 11 or 12, further comprising:

providing a first centralized cache component for the plurality of nodes.

14. The system of claim 13, further comprising:

in response to determining that the requested state is not available in the plurality of cache components of the cache-mesh, loading the requested state from the first centralized cache component.

15. The system of one of claims 11 to 14, wherein the protocol for communication includes a component selected from the group consisting of an identifier of a user, a key identifying a cached value, and the cached value.

16. The system of claim 13, further comprising:

in response to completion of the execution of the execution container on the respective node, sending the end state to the first centralized cache component.

17. The system of one of claims 11 to 16, wherein the plurality of cache components of the cache-mesh include a pool of cache components having a comparable workload type.

18. The system of one of claims 11 to 17, wherein the comparable workload type includes an execution pattern selected from the group consisting of an identical runtime engine, a user information, and an identical programming language.

19. The system of claim 13, further comprising:

synchronizing a second centralized cache component with the first centralized cache component using a replication protocol, wherein the second centralized cache component is maintained in a second remote workload execution environment.

20. A computer program product for execution of a stateless service on a node in a workload execution environment, the computer program product comprising:

one or more computer-readable storage media and program instructions collectively stored on the one or more computer-readable storage media, the program instructions comprising:

program instructions to define at least one workload container for a respective node of a plurality of nodes, wherein the at least one workload container includes a cache component of a cache-mesh, wherein the cache component includes a local memory for the at least one workload container, wherein the cache-mesh includes a plurality of cache components across different nodes of the plurality of nodes, wherein the plurality of cache components includes a protocol for communication between each other;

program instructions to in response to receiving a state request from a stateless requesting service from one of the plurality of cache components of the cache-mesh in an execution container, determine whether a requested state associated with the state request is present in the cache component of a related execution container;

program instructions to in response to a cache miss, broadcast the state request to other cache components of the plurality of cache components of the cache-mesh;

program instructions to determine, by the other cache components of the plurality of cache components of the cache-mesh, whether the requested state is present in respective caches of the other cache components; and

program instructions to in response to any cache component identifying the requested state, send the requested state to the requesting service using the protocol for communication.