

CLAIMS

We claim:

- 5 1. At least one computer readable storage medium comprising a set of executable computer programming instructions for providing enhanced function as a service (FaaS) to a plurality of users, which, when executed by a computing system, cause the computing system to:
- execute a plurality of functions on one or more architectural subsystems in the
10 computing system in response to one or more events received from the plurality of users, the one or more architectural subsystems representing an abstraction of execution environment for the plurality of functions and a plurality of containers associated with the plurality of functions;
- allocate a plurality of computing resources of the computing system to
15 facilitate the execution of the plurality of functions by one or more software and orchestration subsystems in the computing system;
- analyze a plurality of parameters associated with the plurality of functions and a plurality of parameters associated with the plurality of computing resources;
- store the plurality of functions and analysis of the plurality of parameters
20 associated with the plurality of functions and the plurality of computing resources in one or more networking and storage subsystems in the computing system, locations for storing of the plurality of functions and the analysis of the plurality of parameters being selected for enhancing locality between the plurality of functions and the corresponding plurality of computing resources and reducing function execution
25 latency; and
- secure the execution of the plurality of function by one or more security subsystems in the computing system.
2. The at least one computer readable storage medium of claim 1,
30 wherein the executable computer programming instructions, when executed by the computing system, cause the computing system to:
- monitor the execution of one or more functions of the plurality of functions;
- partition one or more computing resources of the computing system into one or more shared resources, each function of the plurality of functions having access to

the one or more shared resource;

provide a scheduling of allocating one or more computing resources for executing the plurality of function, the scheduling being generated based, at least, upon history-based resource scheduling of functions being executed by the computing system;

redirect data of the one or more functions to selected computing devices for execution;

select one or more functions of the plurality of function according to service level parameters associated with the one or more functions;

combine the selected one or more functions into a combined function for execution.

3. The at least one computer readable storage medium of claim 1, wherein the executable computer programming instructions, when executed by the computing system, cause the computing system to:

receive a trigger agent for invoking a second function of the plurality of function, the second function being executed subsequent to a function of the plurality of functions being currently executed;

provide feedback to the trigger agent to indicate readiness for the second function invocation;

in response to a function of the plurality of function being a multi-tenant accelerated function, initiate one or more versions of the function;

provide synchronization among execution actions associated with the executing of the plurality of functions, the execution actions being distributed among a plurality of computing devices and/or containers associated with the plurality of functions; and

trigger the second function invocation in response to a criterion being met.

4. The at least one computer readable storage medium of claim 1, wherein the executable computer programming instructions, when executed by the computing system, cause the computing system to:

identify shared information among two or more functions of the plurality of functions;

analyze whether a function called for execution requires the shared information for execution; and
route the function to an execution environment based on the analysis.

5 5. The at least one computer readable storage medium of claim 1, wherein the executable computer programming instructions, when executed by the computing system, cause the computing system to:

- analyze a function during an initial execution of the function;
- store usage data generated from the analysis of the function as a meta data file
- 10 for subsequent invocation of the function, each invocation of the function generating at least one meta data file for the function;
- reclaim the function based on one or more factors, each of the one or more factors indicating execution status of the function, the execution status indicating whether to reclaim the function; and
- 15 apply a security policy to the function, the security policy associated with the execution environment of the function, one or more function sets of the function being encoded with a same security key.

20 6. The at least one computer readable storage medium of claim 1, wherein the executable computer programming instructions, when executed by the computing system, cause the computing system to:

- identify one or more computing resources needs of a function of the plurality of functions;
- pre-reserve one or more identified computing resources in a container to
- 25 which the function is assigned, each identified computing resource having a resource directory identification;
- apply an accelerator to evaluate a cost function for specific workloads of the function;
- construct data transfer and communication links between the pre-reserved
- 30 computing resources and invocation of the function to reduce data transfer or communication latency; and
- monitor one or more quality of service (QoS) measurements associated with the execution of the function, at least one QoS measurement being defined by a vector.

7. The at least one computer readable storage medium of claim 6, further comprising executable computer programming instructions, when executed by the computing system, cause the computing system to:

5 detect one or more anomalies at runtime of executing the function, the anomalies being detected using dynamic profiling capabilities in the runtime of executing the function;

report results of the detecting of the one or more anomalies to at least one performance analysis tool; and

10 maintain a profile mapping of function based on the dynamic profiling capabilities of the function.

8. The at least one computer readable storage medium of claim 1, wherein the executable computer programming instructions, when executed by the computing system, cause the computing system to:

15 associate resource characteristics with executed functions to generate demand fingerprints of the executed functions at each stage of execution of the functions, the demand fingerprints being associated with parameters of the functions and at least one tenant that invokes the functions, and the demand fingerprints being generated based on application of a trained sequence analytics machine learning model;

20 generate reports on the usage of different resources at multiple stages of execution of the functions; and

allocate resources to the functions for executing the functions based on the generated reports.

25

9. The at least one computer readable storage medium of claim 1, wherein the executable computer programming instructions, when executed by the computing system, cause the computing system to:

30 detect a first call by a first invocation instance of a function that returns a set of fields;

identify a first subset of the set of fields, the first subset being associated with the first invocation instance; and

conduct a first relay of a first response object based on the first subset, the layout of the first response object reordering or reconfiguring the first response object.

5 10. The at least one computer readable storage medium of claim 1, wherein the executable computer programming instructions, when executed by the computing system, cause the computing system to:

 determine a payload for a function of the plurality of functions; and
 place, at compile time, a code that builds the payload onto a call stack in a
10 format that contains information sufficient to make a call, wherein the function is uniformly invocable via the payload as a local instance, a remote instance or hardware.

 11. The at least one computer readable storage medium of claim 1, wherein the executable computer programming instructions, when executed by the
15 computing system, cause the computing system to:

 determine whether capability information associated with a first function of the plurality of functions is valid with respect to a first authentication format, wherein the capability information and the first authentication format correspond to the first
20 function;

 store the capability information to a queue if the capability information is valid with respect to the first authentication format; and

 transfer the capability information to a second function of the plurality of functions in accordance with a second authentication format that corresponds to the
25 second function, wherein the first function and the second function are portable functions, a portable function having a portable identity and being dynamically movable throughout the computing system by a load balancer, and wherein the capability information includes hierarchical encoded inline capabilities and user-level interrupts.

30

 12. The at least one computer readable storage medium of claim 1, wherein the executable computer programming instructions, when executed by the computing system, cause the computing system to:

map a key identifier to a first key, wherein the first key is associated with a first function and a first memory region that is encrypted with the first key;
detect a context switch from the first function to a second function; and
map the key identifier to a second key in response to the context switch,
5 wherein the second key is associated with the second function and a second memory region that is encrypted with the second key,
wherein the first memory region and the second memory region share a linear address range,
wherein the first key is a key for a first target domain and the second key is a
10 key for a second target domain, and wherein one or more of the first target domains or the second target domains are a sub-domain of a trust domain.

13. The at least one computer readable storage medium of claim 1,
wherein the executable computer programming instructions, when executed by the
15 computing system, cause the computing system to:
detect an insufficient number of protection key identifiers with respect to a set of functions of the plurality of functions;
instruct a host computing device to update a page table entry in response to the
insufficient number of protection key identifiers, wherein the set of functions are
20 limited to managed runtime functions; and
enforce an atomic execution of the set of functions.

14. The at least one computer readable storage medium of claim 1,
wherein the executable computer programming instructions, when executed by the
25 computing system, cause the computing system to:
determine whether a software thread is running on a hardware logical thread;
swap a tag into a register when the software thread is running on the hardware
logical thread; and
set at least one of a cache capacity and memory bandwidth for the tag.

30
15. A semiconductor apparatus comprising:
one or more substrates; and

logic coupled to the one or more substrates, wherein the logic is implemented at least partly in one or more of configurable logic or fixed-functionality hardware logic, the logic coupled to the one or more substrates to:

5 execute a plurality of functions on one or more architectural subsystems in the computing system in response to one or more events received from the plurality of users, the one or more architectural subsystems representing an abstraction of execution environment for the plurality of functions and a plurality of containers associated with the plurality of functions;

10 allocate a plurality of computing resources of the computing system to facilitate the execution of the plurality of functions by one or more software and orchestration subsystems in the computing system;

analyze a plurality of parameters associated with the plurality of functions and a plurality of parameters associated with the plurality of computing resources;

15 store the plurality of functions and analysis of the plurality of parameters associated with the plurality of functions and computing resources in one or more networking and storage subsystems in the computing system, locations for storing of the plurality of functions and the analysis of the plurality of parameters being selected for enhancing locality between the plurality of functions and the corresponding plurality of computing resources and reducing function execution latency; and

20 secure the execution of the plurality of function by one or more security subsystems in the computing system.

16. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

25 monitor the execution of one or more functions of the plurality of functions; partition one or more computing resources of the computing system into one or more shared resources, each function of the plurality of functions having access to the one or more shared resource;

30 provide a scheduling of allocating one or more computing resources for executing the plurality of function, the scheduling being generated based, at least, upon history-based resource scheduling of functions being executed by the computing system;

redirect data of the one or more functions to selected computing devices for execution;

select one or more functions of the plurality of function according to service level parameters associated with the one or more functions;

combine the selected one or more functions into a combined function for execution.

5

17. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

receive a trigger agent for invoking a second function of the plurality of function, the second function being executed subsequent to a function of the plurality of functions being currently executed;

provide feedback to the trigger agent to indicate readiness for the second function invocation;

in response to a function of the plurality of function being a multi-tenant accelerated function, initiate one or more versions of the function;

provide synchronization among execution actions associated with the executing of the plurality of functions, the execution actions being distributed among a plurality of computing devices and/or containers associated with the plurality of functions; and

trigger the second function invocation in response to a criterion being met.

20

18. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

identify shared information among two or more functions of the plurality of functions;

analyze whether a function called for execution requires the shared information for execution; and

route the function to an execution environment based on the analysis.

19. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

analyze a function during an initial execution of the function;

store usage data generated from the analysis of the function as a meta data file for subsequent invocation of the function, each invocation of the function generating at least one meta data file for the function;

reclaim the function based on one or more factors, each of the one or more factors indicating execution status of the function, the execution status indicating whether to reclaim the function; and

5 apply a security policy to the function, the security policy associated with the execution environment of the function, one or more function sets of the function being encoded with a same security key.

20. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

10 identify one or more computing resources needs of a function of the plurality of functions;

 pre-reserve one or more identified computing resources in a container to which the function is assigned, each identified computing resource having a resource directory identification;

15 apply an accelerator to evaluate a cost function for specific workloads of the function;

 construct data transfer and communication links between the pre-reserved computing resources and invocation of the function to reduce data transfer or communication latency; and

20 monitor one or more quality of service (QoS) measurements associated with the execution of the function, at least one QoS measurement being defined by a vector.

21. The semiconductor apparatus of claim 20, wherein the logic coupled to the one or more substrates is to:

 detect one or more anomalies at runtime of executing the function, the anomalies being detected using dynamic profiling capabilities in the runtime of executing the function;

30 report results of the detecting of the one or more anomalies to at least one performance analysis tool; and

 maintain a profile mapping of function based on the dynamic profiling capabilities of the function.

22. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

associate resource characteristics with executed functions to generate demand fingerprints of the executed functions at each stage of execution of the functions, the demand fingerprints being associated with parameters of the functions and at least one
5 tenant that invokes the functions, and the demand fingerprints being generated based on application of a trained sequence analytics machine learning model;

generate reports on the usage of different resources at multiple stages of execution of the functions; and

10 allocate resources to the functions for executing the functions based on the generated reports.

23. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

15 detect a first call by a first invocation instance of a function that returns a set of fields;

identify a first subset of the set of fields, the first subset being associated with the first invocation instance; and

20 conduct a first relayout of a first response object based on the first subset, the relayout of the first response object reordering or reconfiguring the first response object.

24. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

25 determine a payload for a function of the plurality of functions; and

place, at compile time, a code that builds the payload onto a call stack in a format that contains information sufficient to make a call, wherein the function is uniformly invocable via the payload as a local instance, a remote instance or hardware.

30

25. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

determine whether capability information associated with a first function of the plurality of functions is valid with respect to a first authentication format, wherein

the capability information and the first authentication format correspond to the first function;

store the capability information to a queue if the capability information is valid with respect to the first authentication format; and

5 transfer the capability information to a second function of the plurality of functions in accordance with a second authentication format that corresponds to the second function, wherein the first function and the second function are portable functions, a portable function having a portable identity and being dynamically movable throughout the computing system by a load balancer, and wherein the
10 capability information includes hierarchical encoded inline capabilities and user-level interrupts.

26. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

15 map a key identifier to a first key, wherein the first key is associated with a first function and a first memory region that is encrypted with the first key;

detect a context switch from the first function to a second function; and

map the key identifier to a second key in response to the context switch,

wherein the second key is associated with the second function and a second

20 memory region that is encrypted with the second key,

wherein the first memory region and the second memory region share a linear address range,

wherein the first key is a key for a first target domain and the second key is a key for a second target domain, and wherein one or more of the first target domains or
25 the second target domains are a sub-domain of a trust domain.

27. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

30 detect an insufficient number of protection key identifiers with respect to a set of functions of the plurality of functions;

instruct a host computing device to update a page table entry in response to the insufficient number of protection key identifiers, wherein the set of functions are limited to managed runtime functions; and

enforce an atomic execution of the set of functions.

28. The semiconductor apparatus of claim 15, wherein the logic coupled to the one or more substrates is to:

- determine whether a software thread is running on a hardware logical thread;
- 5 swap a tag into a register when the software thread is running on the hardware logical thread; and
- set at least one of a cache capacity and memory bandwidth for the tag.

29. A system comprising:

- 10 a memory; and
- a processor coupled to the memory, the processor including logic to execute a plurality of functions on one or more architectural subsystems in the computing system in response to one or more events received from the plurality of users, the one or more architectural subsystems representing an abstraction of
- 15 execution environment for the plurality of functions and a plurality of containers associated with the plurality of functions;
- allocate a plurality of computing resources of the computing system to facilitate the execution of the plurality of functions by one or more software and orchestration subsystems in the computing system;
- 20 analyze a plurality of parameters associated with the plurality of functions and a plurality of parameters associated with the plurality of computing resources;
- storing the plurality of functions and analysis of the plurality of parameters associated with the plurality of functions and the plurality of computing resources in one or more networking and storage subsystems in the computing system, locations
- 25 for storing of the plurality of functions and the analysis of the plurality of parameters being selected for enhancing locality between the plurality of functions and the corresponding plurality of computing resources and reducing function execution latency; and
- secure the execution of the plurality of function by one or more security
- 30 subsystems in the computing system.

30. A method comprising:

- executing a plurality of functions on one or more architectural subsystems in the computing system in response to one or more events received from the plurality of

users, the one or more architectural subsystems representing an abstraction of execution environment for the plurality of functions and a plurality of containers associated with the plurality of functions;

5 allocating a plurality of computing resources of the computing system to facilitate the execution of the plurality of functions by one or more software and orchestration subsystems in the computing system;

analyzing a plurality of parameters associated with the plurality of functions and a plurality of parameters associated with the plurality of computing resources;

10 storing the plurality of functions and analysis of the plurality of parameters associated with the plurality of functions and computing resources in one or more networking and storage subsystems in the computing system, locations for storing of the plurality of functions and the analysis of the plurality of parameters being selected for enhancing locality between the plurality of functions and the corresponding plurality of computing resources and reducing function execution latency; and

15 securing the execution of the plurality of function by one or more security subsystems in the computing system.