

METHOD, SYSTEM, AND COMPUTER PROGRAM PRODUCT FOR SCHEDULING OF FUNCTIONS ON DISTRIBUTED SYSTEMS

BACKGROUND

1. Technical Field

[0001] This disclosure relates generally to serverless computing and, in non-limiting embodiments or aspects, to methods, systems, and computer program products for efficient scheduling of serverless functions on distributed systems.

2. Technical Considerations

[0002] Serverless computing has become popular for applications looking for on-demand scalability, deployment abstraction, and/or unit billing. However, there are criticisms of the design of serverless architectures. One such argument was presented in the paper titled “Serverless Computing: One Step Forward, Two Steps Back” by Hellerstein et.al in 2018, in which the authors presented the following two concerns which are inherent to the design of serverless architecture: 1) Function as a Service (FaaS) is a data-shipping architecture and 2) FaaS stymies distributed computing (e.g., because serverless functions are non-addressable, serverless functions cannot directly communicate with each other and, instead, use expensive (in terms of read/write throughput) storage operations to communicate data.

SUMMARY

[0003] Accordingly, provided are improved methods, systems, and computer program products for scheduling of functions on distributed systems.

[0004] According to non-limiting embodiments or aspects, provided is a method, including: storing, with at least one processor, a mapping between a plurality of data points and a plurality of storage nodes on which the plurality of data points is stored; receiving, with the at least one processor, a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function; determining, with the at least one processor, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored; determining, with the at least one processor, based on distance data associated with a plurality of distances between the at least one storage node and a

plurality of compute nodes, at least one compute node of the plurality of compute nodes to schedule to execute the function; and scheduling, with the at least one processor, the function for execution on the at least one compute node.

[0005] In some non-limiting embodiments or aspects, the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes and a plurality of node health parameters associated with the plurality of compute nodes.

[0006] In some non-limiting embodiments or aspects, the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a tunable parameter.

[0007] In some non-limiting embodiments or aspects, the distance data associated with the plurality of distances between the at least one storage node and a plurality of compute nodes includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes.

[0008] In some non-limiting embodiments or aspects, the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the function during execution of the function, and wherein the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function.

[0009] In some non-limiting embodiments or aspects, the method further includes: executing, with the at least one processor, the function on the at least one compute node, wherein executing the function on the at least one compute node accesses the at least one data point at the at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored.

[0010] In some non-limiting embodiments or aspects, the method further includes: updating, with the at least one processor, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes.

[0011] According to some non-limiting embodiments or aspects, provided is a system including: at least one processor coupled to a memory and configured to: store a mapping between a plurality of data points and a plurality of storage nodes on which

the plurality of data points is stored; receive a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function; determine, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored; determine, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute nodes, at least one compute node of the plurality of compute nodes to schedule to execute the function; and schedule, the function for execution on the at least one compute node.

[0012] In some non-limiting embodiments or aspects, the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes and a plurality of node health parameters associated with the plurality of compute nodes.

[0013] In some non-limiting embodiments or aspects, the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a tunable parameter.

[0014] In some non-limiting embodiments or aspects, the distance data associated with the plurality of distances between the at least one storage node and a plurality of compute nodes includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes.

[0015] In some non-limiting embodiments or aspects, the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the function during execution of the function, and wherein the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function.

[0016] In some non-limiting embodiments or aspects, the at least one processor is further configured to: execute the function on the at least one compute node, wherein executing the function on the at least one compute node accesses the at least one data point at the at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored.

[0017] In some non-limiting embodiments or aspects, the at least one processor is further configured to: update, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes.

[0018] According to some non-limiting embodiments or aspects, provided is a computer program product, including at least one non-transitory computer-readable medium including program instructions that, when executed by at least one processor, cause the at least one processor to: store a mapping between a plurality of data points and a plurality of storage nodes on which the plurality of data points is stored; receive a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function; determine, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored; determine, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute nodes, at least one compute node of the plurality of compute nodes to schedule to execute the function; and schedule, the function for execution on the at least one compute node.

[0019] In some non-limiting embodiments or aspects, the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes and a plurality of node health parameters associated with the plurality of compute nodes.

[0020] In some non-limiting embodiments or aspects, the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a tunable parameter.

[0021] In some non-limiting embodiments or aspects, the distance data associated with the plurality of distances between the at least one storage node and a plurality of compute nodes includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes.

[0022] In some non-limiting embodiments or aspects, the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the

function during execution of the function, and wherein the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function.

[0023] In some non-limiting embodiments or aspects, the program instructions, when executed by the at least one processor, further cause the at least one processor to: execute the function on the at least one compute node, wherein executing the function on the at least one compute node accesses the at least one data point at the at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored; and update, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes.

[0024] Further non-limiting embodiments or aspects are set forth in the following numbered clauses:

[0025] Clause 1. A method, comprising: storing, with at least one processor, a mapping between a plurality of data points and a plurality of storage nodes on which the plurality of data points is stored; receiving, with the at least one processor, a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function; determining, with the at least one processor, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored; determining, with the at least one processor, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute nodes, at least one compute node of the plurality of compute nodes to schedule to execute the function; and scheduling, with the at least one processor, the function for execution on the at least one compute node.

[0026] Clause 2. The method of clause 1, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes and a plurality of node health parameters associated with the plurality of compute nodes.

[0027] Clause 3. The method of clause 1 or 2, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a tunable parameter.

[0028] Clause 4. The method of any of clauses 1-3, wherein the distance data associated with the plurality of distances between the at least one storage node and a plurality of compute nodes includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes.

[0029] Clause 5. The method of any of clauses 1-4, wherein the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the function during execution of the function, and wherein the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function.

[0030] Clause 6. The method of any of clauses 1-5, further comprising: executing, with the at least one processor, the function on the at least one compute node, wherein executing the function on the at least one compute node accesses the at least one data point at the at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored.

[0031] Clause 7. The method of any of clauses 1-6, further comprising: updating, with the at least one processor, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes.

[0032] Clause 8. A system comprising: at least one processor coupled to a memory and configured to: store a mapping between a plurality of data points and a plurality of storage nodes on which the plurality of data points is stored; receive a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function; determine, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored; determine, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute nodes, at least one compute

node of the plurality of compute nodes to schedule to execute the function; and schedule, the function for execution on the at least one compute node.

[0033] Clause 9. The system of clause 8, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes and a plurality of node health parameters associated with the plurality of compute nodes.

[0034] Clause 10. The system of clause 8 or 9, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a tunable parameter.

[0035] Clause 11. The system of any of clauses 8-10, wherein the distance data associated with the plurality of distances between the at least one storage node and a plurality of compute nodes includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes.

[0036] Clause 12. The system of any of clauses 8-11, wherein the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the function during execution of the function, and wherein the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function.

[0037] Clause 13. The system of any of clauses 8-12, wherein the at least one processor is further configured to: execute the function on the at least one compute node, wherein executing the function on the at least one compute node accesses the at least one data point at the at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored.

[0038] Clause 14. The system of any of clauses 8-13, wherein the at least one processor is further configured to: update, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes.

[0039] Clause 15. A computer program product, comprising at least one non-transitory computer-readable medium including program instructions that, when executed by at least one processor, cause the at least one processor to: store a mapping between a plurality of data points and a plurality of storage nodes on which

the plurality of data points is stored; receive a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function; determine, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored; determine, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute nodes, at least one compute node of the plurality of compute nodes to schedule to execute the function; and schedule, the function for execution on the at least one compute node.

[0040] Clause 16. The computer program product of clause 15, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes and a plurality of node health parameters associated with the plurality of compute nodes.

[0041] Clause 17. The computer program product of clause 15 or 16, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a tunable parameter.

[0042] Clause 18. The computer program product of any of clauses 15-17, wherein the distance data associated with the plurality of distances between the at least one storage node and a plurality of compute nodes includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes.

[0043] Clause 19. The computer program product of any of clauses 15-18, wherein the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the function during execution of the function, and wherein the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function.

[0044] Clause 20. The computer program product of any of clauses 15-19, wherein the program instructions, when executed by the at least one processor, further cause the at least one processor to: execute the function on the at least one compute node, wherein executing the function on the at least one compute node accesses the at least one data point at the at least one storage node of the plurality of storage nodes on

which the at least one data point is stored or to be stored; and update, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes.

[0045] These and other features and characteristics of the present disclosure, as well as the methods of operation and functions of the related elements of structures and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only and are not intended as a definition of the limits of the disclosed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0046] Additional advantages and details are explained in greater detail below with reference to the non-limiting, exemplary embodiments that are illustrated in the accompanying schematic figures, in which:

[0047] FIG. 1 is a schematic diagram of a system for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects;

[0048] FIG. 2 is a schematic diagram of example components of one or more devices of FIG. 1, according to some non-limiting embodiments or aspects;

[0049] FIG. 3 is a flow diagram of a method for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects;

[0050] FIG. 4 is a schematic diagram of an implementation of a system for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects;

[0051] FIG. 5 is a schematic diagram of an implementation of a system for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects; and

[0052] FIG. 6 is a schematic diagram of an implementation of a system for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects.

DETAILED DESCRIPTION

[0053] For purposes of the description hereinafter, the terms “end,” “upper,” “lower,” “right,” “left,” “vertical,” “horizontal,” “top,” “bottom,” “lateral,” “longitudinal,” and derivatives thereof shall relate to the embodiments as they are oriented in the drawing figures. However, it is to be understood that the present disclosure may assume various alternative variations and step sequences, except where expressly specified to the contrary. It is also to be understood that the specific devices and processes illustrated in the attached drawings, and described in the following specification, are simply exemplary and non-limiting embodiments or aspects of the disclosed subject matter. Hence, specific dimensions and other physical characteristics related to the embodiments or aspects disclosed herein are not to be considered as limiting.

[0054] Some non-limiting embodiments or aspects are described herein in connection with thresholds. As used herein, satisfying a threshold may refer to a value being greater than the threshold, more than the threshold, higher than the threshold, greater than or equal to the threshold, less than the threshold, fewer than the threshold, lower than the threshold, less than or equal to the threshold, equal to the threshold, etc.

[0055] No aspect, component, element, structure, act, step, function, instruction, and/or the like used herein should be construed as critical or essential unless explicitly described as such. Also, as used herein, the articles “a” and “an” are intended to include one or more items and may be used interchangeably with “one or more” and “at least one.” Furthermore, as used herein, the term “set” is intended to include one or more items (e.g., related items, unrelated items, a combination of related and unrelated items, and/or the like) and may be used interchangeably with “one or more” or “at least one.” Where only one item is intended, the term “one” or similar language is used. Also, as used herein, the terms “has,” “have,” “having,” or the like are intended to be open-ended terms. Further, the phrase “based on” is intended to mean “based at least partially on” unless explicitly stated otherwise. In addition, reference to an action being “based on” a condition may refer to the action being “in response to” the condition. For example, the phrases “based on” and “in response to” may, in some non-limiting embodiments or aspects, refer to a condition for automatically triggering an action (e.g., a specific operation of an electronic device, such as a computing device, a processor, and/or the like).

[0056] As used herein, the term “communication” may refer to the reception, receipt, transmission, transfer, provision, and/or the like of data (e.g., information, signals, messages, instructions, commands, and/or the like). For one unit (e.g., a device, a system, a component of a device or system, combinations thereof, and/or the like) to be in communication with another unit means that the one unit is able to directly or indirectly receive information from and/or transmit information to the other unit. This may refer to a direct or indirect connection (e.g., a direct communication connection, an indirect communication connection, and/or the like) that is wired and/or wireless in nature. Additionally, two units may be in communication with each other even though the information transmitted may be modified, processed, relayed, and/or routed between the first and second unit. For example, a first unit may be in communication with a second unit even though the first unit passively receives information and does not actively transmit information to the second unit. As another example, a first unit may be in communication with a second unit if at least one intermediary unit processes information received from the first unit and communicates the processed information to the second unit. In some non-limiting embodiments or aspects, a message may refer to a network packet (e.g., a data packet and/or the like) that includes data. It will be appreciated that numerous other arrangements are possible.

[0057] As used herein, the term “computing device” may refer to one or more electronic devices configured to process data. A computing device may, in some examples, include the necessary components to receive, process, and output data, such as a processor, a display, a memory, an input device, a network interface, and/or the like. A computing device may be a mobile device. As an example, a mobile device may include a cellular phone (e.g., a smartphone or standard cellular phone), a portable computer, a wearable device (e.g., watches, glasses, lenses, clothing, and/or the like), a personal digital assistant (PDA), and/or other like devices. A computing device may also be a desktop computer or other form of non-mobile computer.

[0058] As used herein, the term “server” may refer to or include one or more computing devices that are operated by or facilitate communication and processing for multiple parties in a network environment, such as the Internet, although it will be appreciated that communication may be facilitated over one or more public or private network environments and that various other arrangements are possible. Further, multiple computing devices (e.g., servers, point-of-sale (POS) devices, mobile

devices, etc.) directly or indirectly communicating in the network environment may constitute a “system.”

[0059] As used herein, the term “system” may refer to one or more computing devices or combinations of computing devices (e.g., processors, servers, client devices, software applications, components of such, and/or the like). Reference to “a device,” “a server,” “a processor,” and/or the like, as used herein, may refer to a previously-recited device, server, or processor that is recited as performing a previous step or function, a different device, server, or processor, and/or a combination of devices, servers, and/or processors. For example, as used in the specification and the claims, a first device, a first server, or a first processor that is recited as performing a first step or a first function may refer to the same or different device, server, or processor recited as performing a second step or a second function.

[0060] As used herein, the term “communication network” may refer to one or more wired and/or wireless networks. For example, a communication network may include a cellular network (e.g., a long-term evolution (LTE®) network, a third generation (3G) network, a fourth generation (4G) network, a fifth generation (5G) network, a code division multiple access (CDMA) network, etc.), a public land mobile network (PLMN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), a telephone network (e.g., the public switched telephone network (PSTN)), a private network, an ad hoc network, an intranet, the Internet, a fiber optic-based network, a cloud computing network, and/or the like, and/or a combination of these or other types of networks.

[0061] Non-limiting embodiments or aspects of the present disclosure may provide methods, systems, and/or computer program products that store a mapping between a plurality of data points and a plurality of storage nodes on which the plurality of data points is stored; receive a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function; determine, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored; determine, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute nodes, at least one compute node of the plurality of compute

nodes to schedule to execute the function; and schedule the function for execution on the at least one compute node.

[0062] In this way, non-limiting embodiments or aspects of the present disclosure may enable ensuring that functions are scheduled for compute as close as possible to storage, which may help address the first concern that Function as a Service (FaaS) is a data-shipping architecture, and/or may enable improved or optimal scheduling of a group of functions which may be a part of a same call stack (e.g., which are called one after the another or in some predefined sequence, etc.), which may help to address the second concern that FaaS stymies distributed computing and/or reduce network overhead, due to allowing for indirect communication between functions resulting in the functions being scheduled closer to each other. Accordingly, non-limiting embodiments or aspects of the present disclosure may re-impose a foundational principle of distributing computing, which is of colocation of compute and storage while ensuring that the flexibility provided by serverless computing is not lost, and/or help developers define a way to ensure that related functions or function calls which are part of the same call stack can be located near to each other, which may reduce the penalty of communication via expensive database reads/writes by reducing network latency through placing functions closer to shared storage nodes.

[0063] FIG. 1 is a schematic diagram of a system 100 for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects. As shown in FIG. 1, system 100 may include function scheduler system 102, storage information node 104, a plurality of compute nodes C1, C2, C3, C4, C5, ... CN, a plurality of storage nodes Storage 1, Storage 2, ... Storage N, and/or a plurality of clients SR1, SR2, ... SRN. Function scheduler system 102, storage information node 104, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, the plurality of storage nodes Storage 1, Storage 2, ... Storage N, and/or the plurality of clients SR1, SR2, ... SRN may interconnect (e.g., establish a connection to communicate, etc.) via wired connections, wireless connections, or a combination of wired and wireless connections.

[0064] Function scheduler system 102 may include one or more devices capable of receiving information and/or data from storage information node 104, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, the plurality of storage nodes Storage 1, Storage 2, ... Storage N, and/or the plurality of clients SR1, SR2, ... SRN (e.g., via a communication network, etc.) and/or communicating information and/or data to

storage information node 104, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, the plurality of storage nodes Storage 1, Storage 2, ... Storage N, and/or the plurality of clients SR1, SR2, ... SRN (e.g., via a communication network, etc.). For example, function scheduler system 102 may include a computing device, such as a server, a group of servers, and/or other like devices.

[0065] Function scheduler system 102 may include a highly available entity that acts as an interface between the plurality of clients SR1, SR2, ... SRN, which may generate requests to execute functions, and the plurality of compute nodes C1, C2, C3, C4, C5, ... CN or distributed clusters on which the requested functions may be executed. Function scheduler system 102 may be responsible for scheduling functions for execution on the plurality of compute nodes C1, C2, C3, C4, C5, ... CN based on input from storage information node 104 and/or the function definition itself. Function scheduler system 102 may also use a tunable parameter α , which impacts how strict a policy is applied for attempting to schedule compute nodes as close as possible to storage nodes to be accessed, to schedule the functions for execution on the plurality of compute nodes C1, C2, C3, C4, C5, ... CN.

[0066] Storage information node 104 may include one or more devices capable of receiving information and/or data from function scheduler system 102, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, the plurality of storage nodes Storage 1, Storage 2, ... Storage N, and/or the plurality of clients SR1, SR2, ... SRN (e.g., via a communication network, etc.) and/or communicating information and/or data to function scheduler system 102, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, the plurality of storage nodes Storage 1, Storage 2, ... Storage N, and/or the plurality of clients SR1, SR2, ... SRN (e.g., via a communication network, etc.). For example, storage information node 104 may include a computing device, such as a server, a group of servers, and/or other like devices. In some non-limiting embodiments or aspects, storage information node 104 may be included in and/or implemented by function scheduler system 102.

[0067] Storage information node 104 may be configured to store storage information of the plurality of storage nodes Storage 1, Storage 2, ... Storage N. For example, storage information node 104 may maintain a mapping between the plurality of storage nodes Storage 1, Storage 2, ... Storage N and a plurality of data points such that when storage information node 104 is queried (e.g., by function scheduler

system 102, etc.), storage information node 104 can provide storage node information that indicates which data point is stored on which storage node.

[0068] The plurality of compute nodes C1, C2, C3, C4, C5, ... CN may include one or more devices capable of receiving information and/or data from function scheduler system 102, storage information node 104, the plurality of storage nodes Storage 1, Storage 2, ... Storage N, and/or the plurality of clients SR1, SR2, ... SRN (e.g., via a communication network, etc.) and/or communicating information and/or data to function scheduler system 102, storage information node 104, the plurality of storage nodes Storage 1, Storage 2, ... Storage N, and/or the plurality of clients SR1, SR2, ... SRN (e.g., via a communication network, etc.). For example, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN may be implemented in a distributed system in which individual nodes of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN may be implemented within a single device and/or system or distributed across multiple devices and/or systems of the distributed system. For example, a node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN may include and/or be implemented by a computing device, such as a server, a group of servers, and/or other like devices. As an example, one or more nodes of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN (e.g., one or more computing devices that implement or are included in the one or more nodes of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, etc.) may be located at a different physical location than one or more other nodes of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN (e.g., one or more other computing devices that implement or are included in the one or more other nodes of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, etc.).

[0069] In some non-limiting embodiments or aspects, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN includes a plurality of heterogeneous nodes. For example, different nodes of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN may include different types of hardware, firmware, or a combination of hardware and software (e.g., different types and/or speeds of random access memory (RAM), different types and/or speeds of processors, etc.).

[0070] The plurality of compute nodes C1, C2, C3, C4, C5, ... CN may be configured to execute functions. For example, a compute node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN may be configured to execute a function assigned for execution on that node by function scheduler system 102.

[0071] The plurality of storage nodes Storage 1, Storage 2, ... Storage N may include one or more devices capable of receiving information and/or data from function scheduler system 102, storage information node 104, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, and/or the plurality of clients SR1, SR2, ... SRN (e.g., via a communication network, etc.) and/or communicating information and/or data to function scheduler system 102, storage information node 104, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, and/or the plurality of clients SR1, SR2, ... SRN (e.g., via a communication network, etc.). For example, the plurality of storage nodes Storage 1, Storage 2, ... Storage N may be implemented in a distributed system in which individual nodes of the plurality of storage nodes Storage 1, Storage 2, ... Storage N may be implemented within a single device and/or system or distributed across multiple devices and/or systems of the distributed system. For example, a node of the plurality of storage nodes Storage 1, Storage 2, ... Storage N may include and/or be implemented by a computing device, such as a server, a group of servers, and/or other like devices. As an example, one or more nodes of the plurality of storage nodes Storage 1, Storage 2, ... Storage N (e.g., one or more computing devices that implements or are included in the one or more nodes of the plurality of storage nodes Storage 1, Storage 2, ... Storage N, etc.) may be located at a different physical location than one or more other nodes of the plurality of storage nodes Storage 1, Storage 2, ... Storage N (e.g., one or more other computing devices that implement or are included in the one or more other nodes of the plurality of storage nodes Storage 1, Storage 2, ... Storage N, etc.).

[0072] In some non-limiting embodiments or aspects, the plurality of storage nodes Storage 1, Storage 2, ... Storage N includes a plurality of heterogeneous nodes. For example, different nodes of the plurality of storage nodes Storage 1, Storage 2, ... Storage N may include different types of hardware, firmware, or a combination of hardware and software (e.g., different types and/or speeds of random access memory (RAM), different types and/or speeds of processors, etc.).

[0073] The plurality of plurality of storage nodes Storage 1, Storage 2, ... Storage N may be configured to store a plurality of data points. For example, a function executed by a compute node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN may access (e.g., read, write, etc.) one or more data points of the plurality of data points at the plurality of plurality of storage nodes Storage 1, Storage 2, ... Storage N.

[0074] A data point may include a discrete unit of information, such as a data block including a sequence of bits or bytes, which may contain some whole number of records, having a maximum length (e.g., a block size, etc.), and/or the like. For example, a data block may include a sequence of data in bits or bytes that may be transferred as a whole. The plurality of data points may include a plurality of different types of data. In some non-limiting embodiments or aspects, the plurality of data points may include transaction data associated with a plurality of transactions processed in an electronic payment network. For example, transaction data may include parameters associated with a transaction, such as an account identifier (e.g., a PAN, etc.), a transaction amount, a transaction date and time, a type of products and/or services associated with the transaction, a conversion rate of currency, a type of currency, a merchant type, a merchant name, a merchant location, a transaction approval (and/or decline) rate, and/or the like.

[0075] The plurality of clients SR1, SR2, ... SRN (e.g., a plurality of client systems, a plurality of client devices, etc.) may include one or more devices capable of receiving information and/or data from function scheduler system 102, storage information node 104, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, and/or the plurality of storage nodes Storage 1, Storage 2, ... Storage N (e.g., via a communication network, etc.) and/or communicating information and/or data to function scheduler system 102, storage information node 104, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, and/or the plurality of storage nodes Storage 1, Storage 2, ... Storage N (e.g., via a communication network, etc.). For example, the plurality of clients SR1, SR2, ... SRN may include a computing device, such as a server, a group of servers, and/or other like devices.

[0076] Each client of the plurality of clients SR1, SR2, ... SRN may be configured to provide a request to execute a function (e.g., a software function, etc.) to function scheduler system 102.

[0077] As further shown in FIG. 1, function scheduler system 102, the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, and/or the plurality of storage nodes Storage 1, Storage 2, ... Storage N may provide or implement a background daemon service "Storage Update" that is configured to monitor reads and writes of data points at the plurality of storage nodes Storage 1, Storage 2, ... Storage N by functions executed by the plurality of compute nodes C1, C2, C3, C4, C5, ... CN and provide an indication or log of the reads and writes to storage information node 104 via function

scheduler system 102 to update, based on the reads and writes, the mapping between the plurality of data points and the plurality of storage nodes Storage 1, Storage 2, ... Storage N, for example, so that function scheduler system 102 can take this information into account while making decisions for future function execution requests.

[0078] In some non-limiting embodiments or aspects, the background daemon service “Storage Update” may provide a plurality of load or capacity factors associated with the plurality of storage nodes Storage 1, Storage 2, ... Storage N. For example, a load or capacity factor of a storage node may include a ratio of a number of data blocks stored at the node to a number of addresses (e.g., a number of storage blocks, a number of memory locations, etc.) within the storage node. As an example, the plurality of load or capacity factors may be used to indicate to function scheduler system 102 to take into account load or capacity factors of individual storage nodes when scheduling functions for execution on the plurality of compute nodes C1, C2, C3, C4, C5, ... CN.

[0079] As further shown in FIG. 1, function scheduler system 102 and/or the plurality of storage nodes Storage 1, Storage 2, ... Storage N may provide or implement a background daemon service “Status Update” that is configured to monitor a health of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN and provide node health data or parameters associated therewith to function scheduler system 102 so that function scheduler system 102 can take that information into account when making decisions for function scheduling, for example, compute nodes which are not healthy or are not performing to an acceptable or optimal capacity should not be loaded with more function execution requests, etc.).

[0080] Node health data or parameters may include and/or function scheduler system 102 may determine, based on node health data, a read latency and/or a write latency including a delay (e.g., an amount of time, etc.) introduced or taken by a compute node in serving of a request for access to a data block at a storage node (e.g., an amount of time used by the compute node to serve the request, an amount of time taken due to compute node processing, etc.). For example, node health data may include and/or function scheduler system 102 may determine, based on node health data, an average read and/or write latency including an average delay (e.g., an average amount of time, etc.) introduced or taken by a compute node to serve of one or more requests for access to one or more data blocks at a storage node (e.g., an average amount of time used by the compute node to serve a plurality of requests

from one or more clients, an average amount of time taken due to compute node processing, etc.). As an example, the node health data associated with the plurality of read and/or write latencies associated with the plurality of compute nodes C1, C2, C3, C4, C5, ... CN may include, for each compute node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, a read and/or write latency including an average amount of time taken by that node to serve one or more requests for access to one or more data blocks at one or more storage nodes.

[0081] Referring now to FIG. 2, shown is a diagram of example components of a device 200 according to non-limiting embodiments. Device 200 may correspond to function scheduler system 102, storage information node 104, one or more compute nodes of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, one or more storage nodes of the plurality of storage nodes Storage 1, Storage 2, ... Storage N, and/or one or more clients of the plurality of clients SR1, SR2, ... SRN, as an example. In some non-limiting embodiments or aspects, such systems or devices may include at least one device 200 and/or at least one component of device 200. The number and arrangement of components shown are provided as an example. In some non-limiting embodiments or aspects, device 200 may include additional components, fewer components, different components, or differently arranged components than those shown. Additionally, or alternatively, a set of components (e.g., one or more components) of device 200 may perform one or more functions described as being performed by another set of components of device 200.

[0082] As shown in FIG. 2, device 200 may include a bus 202, a processor 204, memory 206, a storage component 208, an input component 210, an output component 212, and a communication interface 214. Bus 202 may include a component that permits communication among the components of device 200. In some non-limiting embodiments, processor 204 may be implemented in hardware, firmware, or a combination of hardware and software. For example, processor 204 may include a processor (e.g., a central processing unit (CPU), a graphics processing unit (GPU), an accelerated processing unit (APU), etc.), a microprocessor, a digital signal processor (DSP), and/or any processing component (e.g., a field-programmable gate array (FPGA), an application-specific integrated circuit (ASIC), etc.) that can be programmed to perform a function. Memory 206 may include random access memory (RAM), read only memory (ROM), and/or another type of dynamic or static storage

device (e.g., flash memory, magnetic memory, optical memory, etc.) that stores information and/or instructions for use by processor 204.

[0083] With continued reference to FIG. 2, storage component 208 may store information and/or software related to the operation and use of device 200. For example, storage component 208 may include a hard disk (e.g., a magnetic disk, an optical disk, a magneto-optic disk, a solid-state disk, etc.) and/or another type of computer-readable medium. Input component 210 may include a component that permits device 200 to receive information, such as via user input (e.g., a touch screen display, a keyboard, a keypad, a mouse, a button, a switch, a microphone, etc.). Additionally, or alternatively, input component 210 may include a sensor for sensing information (e.g., a global positioning system (GPS) component, an accelerometer, a gyroscope, an actuator, etc.). Output component 212 may include a component that provides output information from device 200 (e.g., a display, a speaker, one or more light-emitting diodes (LEDs), etc.). Communication interface 214 may include a transceiver-like component (e.g., a transceiver, a separate receiver and transmitter, etc.) that enables device 200 to communicate with other devices, such as via a wired connection, a wireless connection, or a combination of wired and wireless connections. Communication interface 214 may permit device 200 to receive information from another device and/or provide information to another device. For example, communication interface 214 may include an Ethernet interface, an optical interface, a coaxial interface, an infrared interface, a radio frequency (RF) interface, a universal serial bus (USB) interface, a Wi-Fi® interface, a cellular network interface, and/or the like.

[0084] Device 200 may perform one or more processes described herein. Device 200 may perform these processes based on processor 204 executing software instructions stored by a computer-readable medium, such as memory 206 and/or storage component 208. A computer-readable medium may include any non-transitory memory device. A memory device includes memory space located inside of a single physical storage device or memory space spread across multiple physical storage devices. Software instructions may be read into memory 206 and/or storage component 208 from another computer-readable medium or from another device via communication interface 214. When executed, software instructions stored in memory 206 and/or storage component 208 may cause processor 204 to perform one or more processes described herein. Additionally, or alternatively, hardwired circuitry may be

used in place of or in combination with software instructions to perform one or more processes described herein. Thus, embodiments described herein are not limited to any specific combination of hardware circuitry and software. The term “configured to,” as used herein, may refer to an arrangement of software, device(s), and/or hardware for performing and/or enabling one or more functions (e.g., actions, processes, steps of a process, and/or the like). For example, “a processor configured to” may refer to a processor that executes software instructions (e.g., program code) that cause the processor to perform one or more functions.

[0085] Referring now to FIG. 3, shown is a flow diagram for a method for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects. The steps shown in FIG. 3 are for example purposes only. It will be appreciated that additional, fewer, different, and/or a different order of steps may be used in some non-limiting embodiments or aspects. In some non-limiting embodiments or aspects, a step may be automatically performed in response to performance and/or completion of a prior step.

[0086] As shown in FIG. 3, at step 302, method 300 includes storing a mapping between a plurality of data points and a plurality of storage nodes on which the plurality of data points is stored. For example, storage information node 104 may store a mapping between a plurality of data points and the plurality of storage nodes Storage 1, Storage 2, ... Storage N on which the plurality of data points is stored. As an example, function scheduler system 102 may store (and/or update), at storage information node 104, a mapping between a plurality of data points and the plurality of storage nodes Storage 1, Storage 2, ... Storage N on which the plurality of data points is stored. In such an example, storage information node 104 may act as a bookkeeping entity that maintains information about which storage node stores what data (e.g., the mapping, etc.).

[0087] As shown in FIG. 3, at step 304, method 300 includes receiving a request to execute a function. For example, function scheduler system 102 may receive a request to execute a function (e.g., a software function, etc.). As an example, function scheduler system 102 may receive, from a client of the plurality of clients SR1, SR2, ... SRN, a request to execute a function (e.g., F1, etc.). In such an example, the request to execute the function may include at least one identifier (e.g., P1, P2, etc.) associated with at least one data point to be accessed by the function during execution of the function. For example, requests to run/execute functions on the distributed

cluster may be received by function scheduler system 102, and/or a function that requests to be executed on the distributed cluster may declare, before execution, the external data connections (e.g., reads, writes, etc.) that the function plans to make throughout an execution lifetime of the function, by letting function scheduler system 102 know the details of the exact data points that the function plans to access. This information enables function scheduler system 102 to take into consideration the type of data that the function is expected to access and make a scheduling decision based on that information.

[0088] In some non-limiting embodiments or aspects, the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the function during execution of the function, and/or the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function. For example, a request to execute a function may include an indication of at least one storage node to which at least one data point is to be written by the function during execution of the function. For example, the sharing of data access information may not be limited to data blocks that the function reads but may also include information about the writes that the function plans to make, such that any future function execution request which depends on the data written by the previous function can be intelligently scheduled close to the nodes where the previous function wrote the data, thereby helping to counter the second concern noted in the Background section that “FaaS Stymies Distributed Computing”. Because the function may identify the exact read and writes that the function plans to do during its lifetime, function scheduler system 102 may take that information into account to ensure that compute is located as close to storage as possible, which was a foundational principle of distributed computing.

[0089] As shown in FIG. 3, at step 306, method 300 includes determining, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored. For example, function scheduler system 102 may determine, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes Storage 1, Storage 2, ... Storage N on which the at least one data point is stored

or to be stored. As an example, function scheduler system 102 may query, based on the at least one identifier associated with the at least one data point, storage information node 104, and storage information node 104 may return, in response to the query including the at least one identifier, at least one storage node of the plurality of storage nodes Storage 1, Storage 2, ... Storage N on which the at least one data point is stored or to be stored. In such an example, the declared external data connections that the function plans to make throughout an execution lifetime of the function may be used by function scheduler system 102 to schedule the function for execution on at least one compute node. For example, once function scheduler system 102 shares the details of the exact data points that the function plans to access with storage information node 104, storage information node 104 may point out a location or node identifier of the at least one node that contains those data points.

[0090] As shown in FIG. 3, at step 308, method 300 includes determining, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute nodes, at least one compute node of the plurality of compute nodes to schedule to execute the function. For example, function scheduler system 102 may determine, based on distance data associated with a plurality of distances between the at least one storage node and the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, at least one compute node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN to schedule to execute the function.

[0091] In some non-limiting embodiments or aspects, the distance data associated with the plurality of distances between the at least one storage node and the plurality of compute nodes C1, C2, C3, C4, C5, ... CN includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes C1, C2, C3, C4, C5, ... CN. For example, a distance between a compute node and a storage node may be calculated based on a ping latency measured between the compute node and the storage node and may be a relative calculation (e.g., the higher the ping latency between two nodes, the larger is the distance between the two nodes, etc.).

[0092] In some non-limiting embodiments or aspects, the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, a plurality of node health parameters associated with the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, one or more tunable parameters, or any combination thereof. For example, function

scheduler system 102 may determine, based on distance data associated with a plurality of distances between the at least one storage node and the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, a plurality of node capacities associated with the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, a plurality of node health parameters associated with the plurality of compute nodes C1, C2, C3, C4, C5, ... CN, one or more tunable parameters, or any combination thereof, at least one compute node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN to schedule to execute the function. As an example, function scheduler system 102 may determine least one compute node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN to schedule to execute the function $f_s(C_i)$ according to the following Equation:

$$f_s(C_i) = (1 - \alpha) \times \sum_{j=0}^k \frac{\text{dist}(C_i, S_j)}{k} + \beta \times (\text{health}) + \gamma \times (\text{capacity})$$

where C_i is a compute node, S_j is a storage node, α is a first tunable parameter, β is a second tunable parameter, and γ is a third tunable parameter

[0093] Tunable parameter α may set or control a strictness of how close a compute node and a storage should be for function execution. For example, and referring also to FIG. 4, which is a schematic diagram of an implementation 400 of a system for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects, if α is high (e.g., close or equal to 1, etc.), function scheduler system 102 may schedule functions with less or least strictness, which may mean that function scheduler system 102 may not give much weight to the distance between the storage node(s) that function will use and the potential compute node(s) on which the function can run or execute. As an example, if α is equal to 1 as shown in FIG. 4, function scheduler system 102 may use a more or most relaxed requirement to locate compute and store close together, in which function scheduler system 102 may not give any weight to colocation in making a scheduling decision. In such an example, running the distributed cluster with $\alpha = 1$ may be a manifestation of existing behavior of scheduling of serverless functions, which is agnostic to storage information.

[0094] Referring now to FIG. 5, which is a schematic diagram of an implementation 500 of a system for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects, if α is equal to 0.5, function scheduler system 102 may use a medium requirement to colocate compute and storage, such that given a storage node, function scheduler system 102 may schedule functions on any of the

three compute nodes shown in FIG. 5, which is a softer requirement as compared to the implementation 400 of FIG. 4, and which may be helpful for function scheduler system 102 to be more flexible in making scheduling decisions as function scheduler system 102 may have more options to schedule a function and give more weight to other parameters, such as node health and node capacity of the extended set of compute nodes to make a scheduling decision.

[0095] Referring now to FIG. 6, which is a schematic diagram of an implementation 600 of a system for scheduling of functions on distributed systems, according to some non-limiting embodiments or aspects, if α is low (e.g., close or equal to 0, etc.), function scheduler system 102 may schedule functions with more or maximum strictness, which may mean that function scheduler system 102 may give more or maximum weight to the distance between the storage node(s) that the function will use and the potential compute nodes on which the function can run or execute (e.g., to colocation of storage and compute and/or ensuring that the function is scheduled to a node that is as close as possible to the nodes that store the information to be used by the function, etc.). As an example, if α is equal to 0 as shown in FIG. 6, function scheduler system 102 may use a strict requirement to locate compute and storage as close as possible, such that given a storage node, function scheduler system 102 may only schedule functions on any of the two compute nodes shown in FIG. 4 that are closest to that particular storage node. Accordingly, in this most strict scenario in which α is equal to 0, the requirement may violate a server anonymity property of serverless architecture.

[0096] The second tunable parameter β may set or control a weight given to node health information or parameters when scheduling functions, and/or the third tunable parameter γ may set or control a weight given to node capacity when scheduling functions.

[0097] As shown in FIG. 3, at step 310, method 300 includes scheduling the function for execution on the at least one compute node. For example, function scheduler system 102 may schedule the function for execution on the at least one compute node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN.

[0098] As shown in FIG. 3, at step 312, method 300 includes executing the function on the at least one compute node. For example, the at least one compute node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN may execute the function. As

an example, function scheduler system 102 may cause the at least one compute node of the plurality of compute nodes C1, C2, C3, C4, C5, ... CN to execute the function at a scheduled time or in a scheduled order with respect to one or more other functions scheduled for execution at the at least one compute node. In such an example, executing the function on the at least one compute node may access (e.g., read, write, etc.) the at least one data point at the at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored.

[0099] As shown in FIG. 3, at step 314, method 300 includes updating, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes. For example, storage information node 104 may update, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes Storage 1, Storage 2, ... Storage N, the mapping between the plurality of data points and the plurality of storage nodes Storage 1, Storage 2, ... Storage N. As an example, function scheduler system 102 may update, at storage information node 104, the mapping between the plurality of data points and the plurality of storage nodes Storage 1, Storage 2, ... Storage N. In such an example, the background daemon service "Storage Update" may provide function scheduler system 102 with an indication or log of the at least one data point being accessed at the at least one storage node of the plurality of storage nodes Storage 1, Storage 2, ... Storage N (e.g., with the reads and writes performed by the executed function, etc.). In some non-limiting embodiments or aspects, it may be assumed that functions make use of external storage to communicate a processing result.

[0100] Although embodiments have been described in detail for the purpose of illustration, it is to be understood that such detail is solely for that purpose and that the disclosure is not limited to the disclosed embodiments or aspects, but, on the contrary, is intended to cover modifications and equivalent arrangements that are within the spirit and scope of the appended claims. For example, it is to be understood that the present disclosure contemplates that, to the extent possible, one or more features of any embodiment or aspect can be combined with one or more features of any other embodiment or aspect.

What is claimed is:

1. A method, comprising:

storing, with at least one processor, a mapping between a plurality of data points and a plurality of storage nodes on which the plurality of data points is stored;

receiving, with the at least one processor, a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function;

determining, with the at least one processor, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored;

determining, with the at least one processor, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute nodes, at least one compute node of the plurality of compute nodes to schedule to execute the function; and

scheduling, with the at least one processor, the function for execution on the at least one compute node.

2. The method of claim 1, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes and a plurality of node health parameters associated with the plurality of compute nodes.

3. The method of claim 2, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a tunable parameter.

4. The method of claim 1, wherein the distance data associated with the plurality of distances between the at least one storage node and the plurality of

compute nodes includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes.

5. The method of claim 1, wherein the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the function during execution of the function, and wherein the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function.

6. The method of claim 1, further comprising:

executing, with the at least one processor, the function on the at least one compute node, wherein executing the function on the at least one compute node accesses the at least one data point at the at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored.

7. The method of claim 6, further comprising:

updating, with the at least one processor, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes.

8. A system comprising:

at least one processor coupled to a memory and configured to:

store a mapping between a plurality of data points and a plurality of storage nodes on which the plurality of data points is stored;

receive a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function;

determine, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored;

determine, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute

nodes, at least one compute node of the plurality of compute nodes to schedule to execute the function; and

schedule the function for execution on the at least one compute node.

9. The system of claim 8, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes and a plurality of node health parameters associated with the plurality of compute nodes.

10. The system of claim 9, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a tunable parameter.

11. The system of claim 8, wherein the distance data associated with the plurality of distances between the at least one storage node and the plurality of compute nodes includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes.

12. The system of claim 8, wherein the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the function during execution of the function, and wherein the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function.

13. The system of claim 8, wherein the at least one processor is further configured to:

execute the function on the at least one compute node, wherein executing the function on the at least one compute node accesses the at least one data point at the at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored.

14. The system of claim 13, wherein the at least one processor is further configured to:

update, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes.

15. A computer program product, comprising at least one non-transitory computer-readable medium including program instructions that, when executed by at least one processor, cause the at least one processor to:

store a mapping between a plurality of data points and a plurality of storage nodes on which the plurality of data points is stored;

receive a request to execute a function, wherein the request to execute the function includes at least one identifier associated with at least one data point to be accessed by the function during execution of the function;

determine, based on the at least one identifier associated with the at least one data point and the mapping, at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored;

determine, based on distance data associated with a plurality of distances between the at least one storage node and a plurality of compute nodes, at least one compute node of the plurality of compute nodes to schedule to execute the function; and

schedule the function for execution on the at least one compute node.

16. The computer program product of claim 15, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a plurality of node capacities associated with the plurality of compute nodes and a plurality of node health parameters associated with the plurality of compute nodes.

17. The computer program product of claim 16, wherein the at least one compute node of the plurality of compute nodes to schedule to execute the function is further determined based on a tunable parameter.

18. The computer program product of claim 15, wherein the distance data associated with the plurality of distances between the at least one storage node and the plurality of compute nodes includes a plurality of ping latencies between the at least one storage node and the plurality of compute nodes.

19. The computer program product of claim 15, wherein the at least one identifier of at least one data point to be accessed by the function during execution of the function includes a plurality of identifiers of a plurality of data points to be accessed by the function during execution of the function, and wherein the plurality of data points includes one or more data points to be read by the function during execution of the function and one or more data points to be written by the function during execution of the function.

20. The computer program product of claim 15, wherein the program instructions, when executed by the at least one processor, further cause the at least one processor to:

execute the function on the at least one compute node, wherein executing the function on the at least one compute node accesses the at least one data point at the at least one storage node of the plurality of storage nodes on which the at least one data point is stored or to be stored; and

update, based on the at least one data point being accessed at the at least one storage node of the plurality of storage nodes, the mapping between the plurality of data points and the plurality of storage nodes.