

Electronic Acknowledgement Receipt

EFS ID:	32678526
Application Number:	15984939
International Application Number:	
Confirmation Number:	9417
Title of Invention:	SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESING
First Named Inventor/Applicant Name:	Oscar E. Ganteaume
Customer Number:	30734
Filer:	Brendan Edward Clark/Monick Monroe
Filer Authorized By:	Brendan Edward Clark
Attorney Docket Number:	46850.24642
Receipt Date:	21-MAY-2018
Filing Date:	
Time Stamp:	15:13:50
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	\$ 1880
RAM confirmation Number	052218INTEFSW00001859502036
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		046850024642Application.pdf	1935295	yes	64
			c111148c5c54954cf38fd2afe7e22e1ad9851f83		
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Power of Attorney		64	64	
	Application Data Sheet		56	63	
	Drawings-only black and white line drawings		46	55	
	Abstract		45	45	
	Claims		39	44	
	Specification		1	38	
	Warnings:				
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	36804	no	2
			fcd3cec779562313b838c823df6c13c2f8041ec8		
Warnings:					
Information:					
Total Files Size (in bytes):			1972099		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESSING

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of and priority to U.S. Patent Application No. 15/969,839, filed May 3, 2018, the entirety of which is incorporated herein by reference.

TECHNICAL FIELD

[0002] This disclosure relates generally to network management and, more specifically, to utilizing multiple cloud services.

BACKGROUND

[0003] Cloud services providers deliver powerful capabilities which enable users to easily access computing resources at a cost. Processing, storage, content delivery, and other local and network functions benefit from the sharing-based economies of scale produced using cloud technology.

[0004] The complexity required to efficiently utilize cloud services can inhibit some companies and individuals from taking advantage of all the benefits that cloud services provide. Learning the complexity and investing in utilizing a particular cloud service solution also “locks in” users to a particular environment, decreasing flexibility and competition.

[0005] This disclosure is directed to solving one or more of the problems in the existing technology.

SUMMARY

[0006] In an aspect, a cloud service management system, comprises an engine. The engine is configured to receive performance metrics from two or more disparate cloud services. The two or more disparate cloud services include a dedicated solution and a shared solution. The engine is further configured to select one or more target resources, among the two or more disparate cloud services, to run tasks based on the performance metrics, a multiservice load balancing scheme, and task parameters. The engine is also configured to provide a scale-up instruction when resource utilization of the tasks, based on the performance metrics, is above a first utilization threshold, and also provide a scale-down instruction when resource utilization of the tasks, based on the performance metrics, is below a second utilization threshold and a deconfliction check is completed. The engine is further configured, in response to the scale-up instruction, to command the dedicated solution to instantiate a new virtual machine to receive at least a portion of the tasks, or to provide a shared resource request message to the shared solution in preparation for handling at least a portion of the tasks. The engine is further configured to, in response to the scale-down instruction, command the dedicated solution to terminate an existing virtual machine, or provide a shared resource release message to the shared solution.

[0007] In another aspect, a method comprises receiving performance metrics from two or more disparate cloud services. The two or more disparate cloud services include a dedicated solution and a shared solution. The method further comprises selecting one or more target resources, among the two or more disparate cloud services, to run tasks based on the performance metrics, a multiservice load balancing scheme, and task parameters. The method also includes providing a scale-up instruction when resource utilization of the tasks, based on the performance metrics, is above a first utilization threshold, and providing a scale-down instruction when resource utilization of the tasks, based on the performance metrics, is

below a second utilization threshold and a deconfliction check is completed. In response to the scale-up instruction, the method comprises commanding the dedicated solution to instantiate a new virtual machine to receive at least a portion of the tasks, or providing a shared resource request message to the shared solution in preparation for handling at least a portion of the tasks. In response to the scale-down instruction, the method comprises commanding the dedicated solution to terminate an existing virtual machine, or providing a shared resource release message to the shared solution.

[0008] According to yet another aspect, a cloud service management system includes non-transitory computer readable media. The media stores instructions that when executed by a processor perform aspects which receive performance metrics from two or more disparate cloud services. The two or more disparate cloud services include a dedicated solution and a shared solution. The instructions when executed further select one or more target resources, among the two or more disparate cloud services, to run tasks based on the performance metrics, a multiservice load balancing scheme, and task parameters. The instructions when executed further provide a scale-up instruction when resource utilization of the tasks, based on the performance metrics, is above a first utilization threshold, and provide a scale-down instruction when resource utilization of the tasks, based on the performance metrics, is below a second utilization threshold and a deconfliction check is completed. In response to the scale-up instruction, further aspects command the dedicated solution to instantiate a new virtual machine to receive at least a portion of the tasks, or provide a shared resource request message to the shared solution in preparation for handling at least a portion of the tasks. In response to the scale-down instruction, further aspects command the dedicated solution to terminate an existing virtual machine, or provide a shared resource release message to the shared solution.

[0009] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter. Furthermore, the claimed subject matter is not limited to limitations that solve any or all disadvantages noted in any part of this disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] In the following description, for purposes of explanation, numerous specific details are set forth in order to provide an understanding of the variations in implementing the disclosed technology. However, the instant disclosure may take many different forms and should not be construed as limited to the examples set forth herein. Where practical, like numbers refer to like elements throughout.

[0011] Fig. 1A is a representation of an example network.

[0012] Fig. 1B is a representation of an example hardware platform for a network.

[0013] Fig. 1C is a representation of an example arrangement for implementing a cloud computing data processing system.

[0014] Fig. 2A is a method that may be used to implement a cloud computing data processing system.

[0015] Fig. 2B is a method that may be used to manage task running modes in a cloud computing data processing system.

[0016] Fig. 2C is a method that may be used to perform load balancing and auto-scaling in a cloud computing data processing system.

[0017] Fig. 2D is a method that may be used to perform virtual machine image importing in disparate cloud solutions.

[0018] Fig. 3 is a schematic of an example device that may be used to manage scaling and load balancing in a cloud computing data processing system.

[0019] Fig. 4A is an example graphical user interface disclosed herein.

[0020] Fig. 4B is an example graphical user interface disclosed herein.

DETAILED DESCRIPTION

[0021] As discussed above, cloud services provide massive resources to all types of users, but often require specialized knowledge to leverage. Accordingly, aspects herein facilitate the use of cloud services by mitigating its inherent complexity in an unconventional manner. Aspects disclosed allow users to create, modify, remove, and launch tasks without having to know how they are executed in the cloud, and in a manner which allows interoperability between cloud service providers and/or their own servers. This diverges from conventional techniques which requires detailed configuration pursuant to proprietary requirements for each type of cloud service based on the manner in which each type of cloud service initializes resources and handles tasks.

[0022] While utilization of cloud service is increasingly common, routine techniques typically involve use of a single cloud service. After implementing task accomplishment in one cloud service, it would be difficult for many users to understand how to achieve the same effects in a different cloud service. The disclosure provides for “translating” between cloud services by consolidating parameters required by available cloud services in a single database and possessing an engine configured to access the fields of the database to prepare information matching both the form and substance for a given cloud service. This obviates the need to create and implement custom virtual machine controllers as well as create implementations for storage controller interfaces in each service. More, the disclosure facilitates combining multiple disparate cloud services when allocating one or

more tasks using a running mode technique. Operational data can be collected from various virtual machines in different cloud services without complex processes, and the disclosed techniques eliminate the need for dependency on cloud service libraries or tools and other proprietary or custom development requirements.

[0023] As used herein, a cloud service is any service, function, or provider of or for cloud resources used to perform tasks in a cloud environment, *e.g.*, providing capability for Software as a Service (SaaS), Platform as a Service (PaaS), Database as a Service (DBaaS), Infrastructure as a Service (IaaS), *et cetera*. Cloud services may utilize virtualization to variously allocate or partition processing power (*e.g.*, compute resources), storage, and networking services or resources according to the needs of a particular cloud user or customer on-demand. Cloud services discussed herein include dedicated cloud solutions and shared cloud solutions. References to disparate or different cloud services (or similar language) indicate two or more cloud services of different types. This could include arrangements with at least one dedicated cloud solution and at least one shared cloud solution; two dedicated cloud solutions of different implementation (*e.g.*, different operating systems or architectures); two shared cloud solutions of different implementation (*e.g.*, two or more of Microsoft Azure, Google Cloud, Amazon Web Services, Alibaba Cloud, IBM cloud services, Oracle cloud services, Samsung cloud services, CenturyLink Cloud, *et cetera*); and others.

[0024] Dedicated cloud solutions use virtual machines under the control of the party utilizing the cloud service. These virtual machines are instantiated, imaged, and terminated according to the user, who may also own or control the resources associated with such virtual machines. The party sends tasks directly to these virtual machines under their control.

[0025] Shared cloud solutions are those managed by the cloud service provider. Rather than control the virtual machines and available resources, users or customers provide tasks (or information for running tasks) to a gateway endpoint preconfigured to run the tasks

using resources available from the shared cloud solution provider. In one example, Amazon Web Services (AWS) describes such tasks as “lambda functions.”

[0026] To run a task using resources managed by the cloud service, information about the task must be provided to a virtual machine on the cloud service capable of handling the task. This information is called “task parameters” herein. Task parameters may be received in a variety of manners, and may be received as “task input” prior to being organized, formatted, encoded, transformed, *et cetera*, into the parametric arrangement used by cloud computing data processing systems and methods disclosed herein and the virtual machines of disparate cloud services used in conjunction therewith. Task parameters can include common parameters and proprietary parameters. Common parameters include those which are used for two or more distinct cloud services, or which are used by a cloud computing data processing system or method disclosed herein. Proprietary parameters include those which are specific to a particular cloud service. In embodiments, the same parameter can be expressed differently based on encoding or formatting. Accordingly, cloud computing data processing systems and methods disclosed herein can encode, format, translate, convert, or otherwise transform input or information related to a task to a standard field for consistent storage of task parameters and to facilitate tasks which are input in one manner to be available for running on cloud services which accept tasks in a different manner.

[0027] Turning to the drawings, Fig. 1A is a representation of an example network 100. Network 100 may include one or more applications (which in turn may include tasks run on virtual machines (VMs) or virtual network functions (VNFs)) implemented on general purpose hardware, such as in lieu of having dedicated hardware for every network function. That is, general purpose hardware of network 100 may be configured to run applications. In embodiments, general purpose hardware may be combined with special purpose hardware,

within hardware platform 106, embodied as element 105, or distributed elsewhere within a network to which elements of Fig. 1A are communicatively coupled, to achieve particular functionality.

[0028] Each application 102 may use one or more VMs 104 or elements 105 to operate. Each VM 104 may have a VM type that indicates its functionality or role. Tasks may be handled by applications, may be applications themselves, or may be processes involving multiple applications or processes not conventionally considered to be applications (*e.g.*, providing file access). Examples of VMs 104 used in administering the cloud may include gateways (GWs), firewalls (FW), routers, real-time analytics, customer edges (vCEs), provider edges (vPEs), proxies, rendezvous points (RPs) or the like. Other VMs may be provided to provide resources to cloud customers or users. Each element 105 may have an element type that indicates its functionality or role. Examples of elements 105 used in administering the cloud include an ingress point, an egress point, a non-virtualized function, or the like. While specific reference may be made to VMs 104 or groups (which may include one or more elements 105), this is for explanation to show that the deployment plan may not necessarily be limited to virtual components in all implementations. As noted earlier, while VMs are discussed for ease and consistency of explanation, this focus may be substituted by or supplemented with focus on containers. For example, one or more of VMs 104 or elements 105 can be a container. Similarly, various clients can be substituted for or comprise application 102, including but not limited to databases, web servers, media transcoders, other cloud applications, *et cetera*, and any tasks contemplated for allocation herein.

[0029] Each VM 104 may consume various network resources from a hardware platform 106, such as resources 108. For example, resources 108 may include one or more virtual central processing units (vCPUs), memory, or a network interface card (NIC). Resources 108 can be dedicated or commingled in support of one or more VM 104, with such

utilization or assignment being performed dynamically, and need not conform to any particular arrangement (*e.g.*, multiple CPUs can support one VM, multiple VMs can be supported by one CPU, *et cetera*). Various rules can be used in such allocation.

[0030] While Fig. 1A illustrates resources 108 as collectively contained in hardware platform 106, the configuration of hardware platform 106 may be further delineated. Fig. 1B provides an example implementation of hardware platform 106.

[0031] Hardware platform 106 may comprise one or more sites 109. For example, a site 109 may be a room, building, or geographic location in which resources 108 are located. For example, site 109 may be a datacenter. Each site 109 may comprise one or more racks 110. In an aspect, rack 110 may refer to the physical housing or platform for multiple servers or other network equipment. In an aspect, rack 110 may also refer to the underlying network equipment. Each rack 110 may include one or more servers 112. Server 112 may comprise general purpose computer hardware or a computer. In an aspect, rack 110 may comprise a metal rack, and servers 112 of rack 110 may comprise blade servers that are physically mounted in or on rack 110.

[0032] Each server 112 may include one or more network resources 108, as illustrated. Servers 112 may be communicatively coupled together (not shown) in any combination or arrangement. For example, all servers 112 within a given site 109 or rack 110 may be communicatively coupled. As another example, servers 112 in different racks 110 may be communicatively coupled. Additionally or alternatively, racks 110 may be communicatively coupled together (not shown) in any combination or arrangement.

[0033] The characteristics of each site 109, rack 110, and server 112 may differ. For example, the number of racks 110 within two sites 109 may vary, or the number of servers 112 within two racks 110 may vary. Additionally or alternatively, the type or number of resources 108 within each server 112 may vary. In an aspect, rack 110 may be used to group

servers 112 with the same resource characteristics. In another aspect, servers 112 within the same rack 110 may have different resource characteristics.

[0034] A single application 102 may include many functional components (*e.g.*, VMs 104 and elements 105). These components may have dependencies upon each other and inter-communication patterns with certain quality of service (QoS) requirements, such as locality, high availability, and security. Consequently, placement decisions—that is, decisions on how (and where) to implement VMs 104 and other elements 105 within hardware platform 106—may be based on all VMs 104 in which the components of application 102 run, including the dependencies of those VMs 104, holistically.

[0035] Such QoS requirements may be domain or application specific. Thus, a deployment plan for assigning resources 108 to VMs 104 of an application may depend upon certain limitations and requirements of both network 100 and application 102, such as the QoS requirements of a given application 102 and the underlying infrastructure of network 100. As all of this information is not typically accessible to both the application provider (that may not have access to network infrastructure details) and the cloud provider (that may not have access to the functionality of application 102), an abstraction may be used to create a deployment plan for application 102, where creation of the deployment plan can occur without knowledge or consideration of the specific infrastructure information.

[0036] A deployment plan may assign VMs 104 to particular resources 108 in accordance with one or more rules in order to account for the requirements of application 102 supported by such VMs 104. These rules may be based on abstracting the requirements of application 102, such as by leveraging the application provider's knowledge on its application 102 to yield a concise and flexible representation of the locality, availability, and security requirements of application 102 without needing to capture the specifics of the cloud infrastructure underlying network 100. The deployment plan may be based on one or more

affinity rules, diversity (or anti-affinity) rules, exclusivity rules, or pipe rules. The deployment plan may further be based on nesting groupings (*e.g.*, rules or sets of VMs 104). For example, the abstraction may provide for certain VMs 104 to be grouped together, so that rules may be applied to groups of VMs 104 or to individual VMs 104. A group may include one or more VMs 104, or other elements 105, such as ingress points, or the like. For example, Fig. 1A shows two example groups 107.

[0037] Figs. 1A and 1B provide an environment for systems described herein, which can include host systems, guest systems, or orchestration systems. Supported in such an environment, or other cloud environments, Fig. 1C illustrates an example cloud arrangement 150 for implementing automated modification of virtualized functions as supported by environments like those of Figs. 1A and 1B. Cloud arrangement 150 includes cloud computing data processing system 152. Cloud computing data processing system 152 includes frontend 154 and backend 164. While illustrated as separate components, frontend 154 and backend 164 may be merged as a single component, or subcomponents may be distributed between frontend 154 and backend 164 in arrangements other than that illustrated, without departing from the scope or spirit of the innovation. Further, cloud computing data processing system 152 can be a locally hosted, run in one or more of dedicated solutions 182 and/or shared solutions 188, run in combinations utilizing both local and remote implementations.

[0038] Cloud 106' may be implemented on hardware 106 (and, in embodiments, similar hardware which is logically or physically separate from hardware 106). Cloud 106' includes a variety of cloud solutions including one or more dedicated solutions 182 and one or more shared solutions 188. Each of dedicated solutions 182 include a plurality of virtual machines 184 and one or more agents 186 for providing monitoring and control of the virtual machines 184. Agents 186 can be deployed to VMs by, *e.g.*, orchestration component 174,

run mode component 168, and/or other components. Shared solutions 188 include a plurality of virtual machines 190 and one or more gateways to provide communication between cloud computing data processing system 152 and virtual machines 190 (and/or shared solutions 188 generally).

[0039] Cloud computing data processing system 152 can use a broker to communicate with agents 186. Various uniform resource locators (URLs) can be used to identify endpoints for communications, and backend 164 (or an engine containing some or all thereof) can have a unique identifier or queue to facilitate routing. Agents 186, which are services for integration with cloud computing data processing system 152, run on each virtual machine in dedicated solutions 182, and execute tasks assigned to the virtual machine and provide reporting and metrics to cloud computing data processing system 152 (*e.g., via* analytics component 176. Proprietary parameters associated with tasks for running said tasks on dedicated solutions 182 can include information for reaching particular agents 186 as well as details required by the respective arrangement of visual machines 184.

[0040] For shared solutions 188, in which cloud computing data processing system 152 and/or administrators thereof do not own or manage portions of the cloud solution, gateways 192 are used to submit tasks for handling by the cloud resources of shared solutions 188. Proprietary parameters associated with tasks for running said tasks on shared solutions 188 can include information for reaching particular gateways 192 as well as details required by the respective arrangement of virtual machines 190.

[0041] When utilizing cloud computing data processing system 152, users 178 can interact with interface 158 or other portions of the system. Applications 180 are shown interacting with analytics component 176. In various embodiments, users 178 can interact with other elements or portions of cloud computing data processing system 152 and/or cloud 106', and applications 180 can interact with various elements or portions of cloud computing

data processing system 152 and/or cloud 106'. More, applications 180 may be hosted by one or both of dedicated solution 182 and shared solution 188 without departing from the scope or spirit of the innovation.

[0042] In the illustrated embodiment, frontend 154 includes interface 158, task repository 160, and visualization component 162. Interface 158 is configured to receive task input including the plurality of task parameters. Task input includes information about tasks to be managed by cloud computing data processing system 152, stored in task repository 160 then utilized by backend 164 to run the tasks using one or more cloud services. Task input can be received as manually entered by a user, in a semi-automatic fashion (*e.g.*, user selection based on automatically populated fields), or automatically (*e.g.*, recognize tasks and import without user entry or selection after the task is identified). In embodiments, interface 158 (or other elements) may be configured to encode the task input to a local format prior to storing the plurality of task parameters in the task repository. In embodiments, the interface can be used for user commands before, during, or after tasks are run, and can be configured to receive, *e.g.*, a task instruction. The task instruction may cause transmission of a communication configured to run a task on at least one of the two or more cloud services. Such a communication includes service parameters required by the at least one of the two or more cloud services to run the task. In alternative or complementary embodiments, the user interface may be configured to receive user input overriding at least one of the plurality of task parameters (*e.g.*, selecting a particular parameter value, running mode, *et cetera*).

[0043] In embodiments utilizing running modes as discussed herein, interface 158 may be a graphical user interface on a display configured to receive a running mode selection, wherein the running mode selection indicates the running mode for allocating the tasks. Running modes for tasks as well as performance metrics of various cloud services, virtual machines, and tasks can also be displayed. Visualization component 162 can work in

conjunction with interface 158 to provide visual representations and infographics depicting the activity and performance of cloud computing data processing system 152 as well as the various cloud services and their VMs. In embodiments, graphical user interface 158 can display a dynamic webpage possessing information describing cloud computing data processing system 152 and related cloud solutions.

[0044] In alternative or complementary embodiments, interface 158 can include one or more Application Programming Interfaces (APIs) that assist task generation and/or importation.

[0045] Interface 158, implemented as a graphical user interface, can include widgets to create, modify, and remove tasks in the repository, as well as display all tasks. Interface 158 can further include a thread drawing canvas to link tasks to create threads or chains of tasks. Interface 158 can also include a thread running control panel allowing users to control (*e.g.*, start, pause, stop, command, interact with) tasks or chains of tasks. Interface 158 can also include elements for parameter definitions beyond the basic task, such as arguments for functions or aspects supporting environment-specific setup. In embodiments, interface 158 also facilitates defining custom metrics for collection by analytics component 176.

[0046] Task repository 160 is configured to store a plurality of task parameters after being received from interface 158. Task repository 160 may be pre-arranged to contain task parameters for one or more tasks without the use of interface 158. The plurality of task parameters are stored in fields associated with task entries in task repository 160. Fields can be automatically populated with default values, or users can be prompted for input, where task input does not satisfy all fields.

[0047] Task parameters can be classified as common parameters and proprietary parameters. Common parameters are those used by multiple or all cloud services to receive and handle tasks, or those which are used by at least one cloud service and the cloud

computing data processing system 152. Proprietary parameters are those which are service-specific (*e.g.*, particular information required for Amazon Web Services to receive a task request which is not required by CenturyLink Cloud) and not used outside of a particular cloud implementation. In embodiments, task repository 160 can store only common parameters, and cloud computing data processing system 152 can encode, format, derive, transform, *et cetera*, the common parameters to satisfy the requirements to provide the tasks to any cloud service.

[0048] Task repository 160 (or other elements of cloud computing data processing system 152) may be configured to link task entries to create linked tasks or threads of tasks for simultaneous, series, or other execution.

[0049] Task repository 160, or other storage, may also store information relating to one or more of dedicated solutions 182 and/or shared solutions 188. Such information can include details not directly related to tasks that still enable cloud computing data processing system 152 to interact with and manage dedicated solutions 182 and shared solutions 188 (for, *e.g.*, load balancing, auto-scaling, collection of metrics, deconflicting running modes). More, task repository 160 or other storage may maintain parameters specific to cloud computing data processing system 152.

[0050] In the illustrated embodiment, backend 164 includes configuration component 166, run mode component 168, auto-scaling component 170, load balancing component 172, orchestration component 174, and analytics component 176. In embodiments, the subcomponents of backend 164 (with or without subcomponents of frontend 154) can be consolidated into an engine. More, even where functional components are defined in distinct modules, it is understood that several may be codependent or interact in a unified fashion (*e.g.*, load balancing component 172 and auto-scaling component 170, configuration component 166 and orchestration component 174).

[0051] Configuration component 166 can be used to configure task requests and other interaction between cloud computing data processing system 152 and the cloud services it leverages or manages including dedicated solutions 182 and shared solutions 188. Each solution may define its own techniques for communications or require particular information, or particularly formatted information, for use. Configuration component 166 ensures that information sent to each solution accords with its requirements, and that traffic from solutions to cloud computing data processing system 152 is understood. In embodiments, configuration component 166 can be used for encoding, transformation, translation, *et cetera*, of task information or other information to standardize such information in accordance with the arrangement of task repository 160 or other components of cloud computing data processing system 152.

[0052] In embodiments, configuration component 166 receives or builds (*e.g.*, collects information and/or encodes) JSON files containing data describing how resources (including VMs 184 in dedicated solutions 182) are created and/or tasks assigned. These JSON files can be task entries and/or other cloud solution information, and can be stored in task repository 160. User customizations can also be included. JSON files can include, *e.g.*, a class configuration including a connection classname (which identifies the broker), a tasking classname (which includes rules for task allocation and certain aspects of load balancing and auto-scaling), a resource classname (which includes rules for orchestration for starting, stopping, and configuring VMs), a storage classname (which includes rules and locations for storage); a logging configuration including a logging configuration file (determines a verbosity level for different components of cloud computing data processing system 152; can be a log4j file), logs directory (log storage location(s)), resource time check (time in seconds to monitor resource loading), and do not terminate list (VMs which should not be terminated regardless of resource load); a VM configuration file (*e.g.*, select and open

file controls for providing and editing JSON files; VMs can read changes without restarting); and a broker configuration including a broker's URL and engine identifier (*e.g.*, identifying backend 164). Addresses can be provided for brokers, configuration information, and other aspects to assist with locating different assets or components of cloud computing data processing system 152 or cloud services with which it interacts.

[0053] Configuration component 166 can also include system storage 194. System storage 194 can store various information or data for further configuration. In an embodiment, system storage 194 can store an image database configured to store a virtual machine image in a stored image format. The image database can alternatively exist in its own storage separate from configuration component 166, or elsewhere in frontend 154, backend 164, cloud 106', *et cetera*. Configuration component 166 or other components of backend 164 can be configured to convert the virtual machine image from the stored image format (*e.g.*, standard image format employed by the image database) to a deployed image format that conforms to formatting for one of the two or more disparate cloud services (*e.g.*, different formats or related data used by AEM, Google, or proprietary dedicated solution, *et cetera*). Configuration component 166, orchestration component 174, or other components can thereafter deploy the virtual machine image in the deployed image format to a virtual machine instance of the one of the two or more disparate cloud services. In embodiments, the stored image format may correspond to a format used by one or more cloud services, and can accordingly be deployed without conversion. For dedicated solutions, agents can be utilized to receive VM images or instruct instantiation of VMs according to images, and for shared solutions images or instructions to instantiate images can be provided to gateways associated with a given shared solution. In embodiments, interface 158, configuration component 166, and/or other components of cloud computing data processing system 152 can receive a virtual machine image, or generate an image of an instantiated virtual machine, and store the virtual

machine image in an image database. In embodiments a received image can be converted from a native image format to the stored image format before storing in the image database.

[0054] Orchestration component 174 provides instructions (*e.g.*, to dedicated solutions 182) or requests (*e.g.*, to shared solutions 188) based on determinations made by cloud computing data processing system 152 or received from users 178. Orchestration component 174 can interact with agents 186 or instantiate new VMs 184 including agents 186. Orchestration component 174 can also interact with gateways 192 to provide requests to shared solution 188.

[0055] In embodiments, configuration component 166 and orchestration component 174 can be used to assign tasks to one or more VMs 184 of the dedicated solutions 182 and/or one or more shared solutions 188. Where assignment is to one or more dedicated solutions 182, dedicated parameters, matching those required for task assignment to each dedicated solution, can be sourced from task repository 160 and appropriately encoded (*e.g.*, to a dedicated format for a given dedicated solution) for handling by one or more of agents 186. Similarly, where assignment is to one or more shared solutions 188, shared parameters, matching those required for task assignment to each shared solution, can be sourced from task repository 160 and appropriately encoded (*e.g.*, to a shared format for a given shared solution) for receipt by one or more of gateways 192. Orchestrator 174 or another component can provide these parameters, as separate parameters or as a consolidated instruction/request. As discussed herein, the target for task allocation may vary based on load balancing, auto scaling, or other techniques.

[0056] Run mode component 168 allows management of task allocation between dedicated solutions 182 and shared solutions 188. Tasks can be allocated according to a running mode implemented by run mode component 168. Running modes include a dedicated mode configured to direct the tasks to one or more dedicated solutions 182, a

serverless mode configured to direct the tasks to one or more shared solutions 188, and a hybrid mode configured to direct the tasks to a combination of the dedicated solutions 182 and the shared solutions 188. Based on the selected running mode, orchestrator component 174 can direct one or more agents 186 associated with one or more virtual machines 184 of the dedicated solutions 182 to execute the tasks if the tasks are allocated according to the dedicated mode or the hybrid mode, or provide task parameters to one or more gateways 192 associated with the shared solutions 188 if the tasks are allocated according to the serverless mode or the hybrid mode. Default running mode(s) can exist which can be override-able by user input or contextual analysis. Running modes can utilize disparate services, assigning tasks indicated for serverless or hybrid handling to multiple disparate services among shared solutions 188, and/or assigning tasks indicated for dedicated or hybrid handling to multiple disparate dedicated solutions 182.

[0057] In embodiments, running mode can be selected by a user interacting with interface 158. In embodiments, running modes can be selected by cloud computing data processing system 152. Where selected by system 152, analytics component 176 can determine costs associated with the tasks based on the dedicated mode, the serverless mode, and the hybrid mode. Dedicated solutions 182 may have certain fixed costs associated with fixed resources, while shared solutions 188 may include greater scalability but costs which increase with resource utilization or prioritization. Accordingly, running modes can be selected to meet cost goals (*e.g.*, cost caps, optimization, cost spreading between vendors, minimum purchase amounts). Similarly, analytics component 176 can be configured to determine (or project) performance or resource consumption associated with the tasks based on the dedicated mode, the serverless mode, and the hybrid mode. Based on necessary service levels, resource caps, urgency, *et cetera*, running mode can be selected based on performance or resource consumption.

[0058] In embodiments, orchestration component 174 can be configured to start or stop (*e.g.*, instantiate or terminate, allocate or deallocate) one or more of VMs 184 based on the dedicated mode or the hybrid mode.

[0059] Selection of the running mode, and more general allocation performed by orchestration component 174, can also be driven by constraints associated with particular solutions. For example, certain cloud services include a timeout period after which a task is terminated whether or not the task was completed. Accordingly, based on task parameters or other information, a running mode can be selected, or more general task allocation made, to avoid assigning tasks which may exceed certain services' timeouts to those services. Other solution constraints, particularly in shared solutions 188, can be built into rules associated with run mode component 168, auto-scaling component 170, load balancing component 172, *et cetera*.

[0060] Auto-scaling component 170, in conjunction with load balancing component 172, creates efficiency between performance and economy by making resources available for tasks based on overall task demand. Auto-scaling component is configured to provide a scale-up instruction when resource utilization of the tasks, based on performance metrics (as developed from, *e.g.*, analytics component 176), is above a first utilization threshold. The first utilization threshold can be a percentage utilization for one or more resources, an absolute amount of resources, a dynamic threshold based on demand projections (*e.g.*, whether an identified task or standard deviation from existing demand can be supported), expected termination times for currently running tasks, *et cetera*, and can be based on context (*e.g.*, based on the number of VMs available, number of tasks running, historical demand; where historical demand is analyzed cloud computing data processing system 152 can utilize pattern recognition, artificial intelligence, or machine learning to develop models or projections). Auto-scaling component is also configured to provide a scale-down instruction

when resource utilization of the tasks, based on the performance metrics, is below a second utilization threshold and a deconfliction check is completed. Once again, the threshold may be static or dynamic and depends on whether resources can be de-allocated without impacting running or anticipated tasks.

[0061] A deconfliction check is conducted to avoid scaling down one of VMs 184 or releasing resources from shared solutions 188 prior to task completion on a respective platform. In embodiments, orchestration component 174 can avoid further task assignment where one of VMs 184 and/or resources of shared solutions 188 are indicated for scaling down. Because embodiments of auto-scaling component 170 may look at overall resource utilization to determine scaling-up or scaling-down instructions, a lower-level inspection of resources to be released can be conducted to ensure no disruption occurs prior to their release.

[0062] Auto-scaling component 170 can provide consistent auto-scaling regardless of cloud solution utilized. Configuration elements for auto-scaling can be ported (*e.g.*, by configuration component 166) between solutions. Such configuration elements can include custom metrics, roles (*e.g.*, permissions for making cloud service requests), launch configuration (*e.g.*, imaging for scaled-up VMs or resources), auto-scaling group definitions, auto-scaling alarms, and auto-scaling policies.

[0063] Based on the instructions, orchestration component 174 is configured to, in response to the scale-up instruction, command the dedicated solution to instantiate a new virtual machine to receive at least a portion of the tasks, or provide a scale-up message (or another message to request additional resources; such messages may be proprietary to one or more of shared solutions 188) to one or more shared solutions 188 in preparation for handling at least a portion of the tasks. Orchestration component 174 is also configured to, in response to the scale-down instruction, command the dedicated solution to terminate an existing virtual

machine, or provide a scale-down message (or another message to release resources; such messages may be proprietary to one or more of shared solutions 188) to the shared solution.

[0064] Scale-up or scale-down instructions can further be based on a number of free agents. In embodiments, cloud computing data processing system 152 can maintain a number of free agents (*e.g.*, minimum or maximum) as an absolute number, resource amount, or based on context such as current or projected traffic or the standard deviation of demand (*e.g.*, more free agents for times or sessions during which demand fluctuates more significantly). Free agents are specific agents among agents 186 associated with VMs 184 that are not presently assigned to any session. These free-agents can be used to satisfy the need, based on a current, requested, or projected session, for a new virtual machine. Free agents allow cloud computing data processing system 152 to have one or more VMs available at any time to provide responsive flexibility to increased demand. When a free agent is utilized (*e.g.*, when demand increases), a scale-up message can be provided to satisfy, *e.g.*, a minimum number of free agents running. However, excess free agents, which useful for minimizing latency, can increase costs; accordingly, a number of free agents can be used in scale-up or scale-down decisions to proactively maintain flexibility and responsiveness without unnecessarily increasing cost.

[0065] Scaling can include rules for observing various constraints or seeking to drive performance or other variables to particular levels. In an embodiment, cost approach rules can be used in conjunction with auto-scaling component 170. For example, performance metrics can include a dedicated solution virtual machine cost associated with one of dedication solutions 182 and a shared solution virtual machine cost (or shared solution resource cost) associated with one of shared solutions 188. Such costs can be dynamic or vary by task or context. In embodiments, a scale-up instruction commands the dedicated solution to instantiate a new virtual machine when the dedicated solution virtual machine cost

is less than the shared solution virtual machine cost or provides the scale-up message (which can be proprietary to the particular shared solution) to the shared solution when the shared solution virtual machine cost is less than the dedicated solution virtual machine cost.

Similarly, a scale-down instruction commands the dedicated solution to terminate the existing virtual machine when the dedicated solution virtual machine cost is greater than the shared solution virtual machine cost or provides the scale-down message (which can be proprietary to the particular shared solution) when the shared solution virtual machine cost is greater than the dedicated solution virtual machine cost.

[0066] Auto-scaling component 170 and related rules significantly ease the burden of establishing auto-scaling in cloud environments. For example, a single implementation of auto-scaling in a given cloud environment can involve significant manual programming requiring creation of custom metrics, creation of roles, creation of launch configuration, creation or identification of auto-scaling groups, creation of scaling alarms to scale up or down, adding scaling policies to scale up or down, *et cetera*. Auto-scaling component utilizes a resource view of cloud 106' across multiple cloud services to manage assigned resources or free agents in those multiple cloud services based on task parameters and information relating to dedicated solution 182 and shared solutions 188.

[0067] Load balancing component 172, based on resources available, determines where to assign tasks among dedicated solutions 182 and shared solutions 188. Load balancing component 172 is configured to receive performance metrics from a plurality of virtual machines which can be associated with disparate cloud services. Load balancing component 172 is further configured to select one or more target virtual machines among VMs 184 and/or VMs 190 (or gateways 192, or more generally shared solutions 188, where particular VMs 190 cannot be selected) to run tasks based on the performance metrics (as

discerned by, *e.g.*, analytics component 176), a multiservice load balancing scheme, and task parameters.

[0068] Load balancing schemes can be based on a variety of rules. In an embodiment, a load balancing scheme can be based on CPU settings associated with the tasks. CPU settings can be task parameters, historical values, or user-provided values. In various embodiments, a first CPU setting provides that the target virtual machine employs a custom task allocation algorithm; a second CPU setting provides that the target virtual machine has CPU availability greater than or equal to the second CPU setting; and/or a third CPU setting provides that the target virtual machine is dedicated to the one of the tasks having the third CPU setting, and wherein the target virtual machine remains dedicated until the one of the tasks is complete. In one or more of the foregoing embodiments, the first CPU setting can be a minimum CPU setting (*e.g.*, 0%), a second CPU setting can be a value greater than zero but less than a maximum CPU setting (*e.g.*, 1% to 99%), and a third setting can be a maximum CPU setting (*e.g.*, 100%).

[0069] In embodiments, a task allocation algorithm of the load balancing scheme (*e.g.*, a custom load balancing algorithm) may include a class. The class includes a routine, and the routine includes aspects to define a configuration set, receive task objects associated with the tasks and a list of including the plurality of virtual machines to assign the tasks, allocate resources, and deallocate resources. In this manner, load balancing and auto-scaling work complementarily. In embodiments, a default task allocation algorithm can assign a task to a virtual machine having the lowest number of tasks.

[0070] As described above, analytics component 176 is configured to monitor performance metrics of the two or more cloud services including the dedicated solution and the shared solution. This is useful as existing cloud solutions do not provide performance metrics in a uniform manner, and may require complicated development efforts to create

services for observing and reporting performance metrics. Cloud computing data processing system 152 automates this process for dedicated solutions 182 and shared solutions 188.

Analytics component 176 can retrieve performance metrics from shared solutions 188 using a routine proprietary to the shared solution and can retrieve performance metrics from agents 186 associated with VMs 184 of dedicated solutions 182.

[0071] In embodiments, analytics component 176 can use recent performance, stored historical performance, or processing requests to discern projected performance. The performance metrics and associated resource utilization can be forward-looking, incorporating the projected performance in task allocation decisions, load balancing, auto-scaling, *et cetera*. Load balancing can alternatively or complementarily be performed based on VM resource loading, first fit, number of tasks, *et cetera*. In embodiments, task parameters can include a load balancing scheme to be used, or applications can select their own load balancing schemes. Custom load balancing schemes can be defined by users or indicated by applications or particular use cases (which are utilized when, *e.g.*, when the CPU setting is set to zero).

[0072] Custom load balancing schemes can take a variety of forms. Examples can include a limit on one or both of a number of tasks handled by a VM or a CPU load threshold (*e.g.*, allow up to 5 tasks or 75% CPU utilization but never above either); a limit on a number of virtual machines (*e.g.*, 5 VMs per session); running related tasks on the same VM to expedite processing (*e.g.*, where similar information used in memory); and others.

[0073] Analytics component 176 can provide data to visualization component 162 to provide visual feedback related to performance, utilization, task status and running times, *et cetera*. Such data can be arranged at various levels of granularity: enterprise level including all resources of cloud 106', solution-level looking at one or more of dedicated solutions 182 and/or shared solutions 188, VM-level or resource-level, task or application

level, user level, *et cetera*. Visualization can include a system view describing tasks in terms of associated VMs, associated tasks, average and/or realtime CPU or memory utilization; session views describing instances, numbers running and completed, numbers failed, as well as resource utilization; and/or an instance view describing tasks, their states, start and finish times, and resource utilization. These can be provided in graphs or charts, which can be color coded to represent different values (*e.g.*, different resources).

[0074] Backend 164 can maintain a task queue for tasks not yet in execution. Tasks run can include any number of tasks, including specialized docker containers or particular VM images. In embodiments, task repository 160 or other storage can include an image database maintaining images for various VMs to be instantiated as standalone tasks or to support running of particular tasks.

[0075] Cloud arrangement 150 is provided as an example of an architecture for utilizing and unifying multiple disparate cloud services. As noted above, while certain components are elements are illustrated in a particular manner to assist with explanation of their operation, it is understood that alternative arrangements can be utilized to achieve similar processing, communication, and management without departing from the scope or spirit of the innovation.

[0076] As can be understood, cloud computing data processing system 152 provides a level of construction which can interact with a variety of disparate cloud systems using a common scheme, providing the same results on different cloud systems without the need for multiple disparate control mechanisms and technical proficiencies. Implementation patterns or scripts can be used to perform actions (*e.g.*, instantiate a VM according to an image) in disparate systems (*e.g.*, AEM and/or Google) without any different action by a human or computerized user, client, administrator, service, *et cetera*. Cloud computing data processing system 152 includes a variety of default configurations to work without customization or

intensive programming, but includes the capability for customization where sophisticated users, clients, administrators, services, *et cetera*, wish to modify operation for their particular uses.

[0077] Continuing, FIGS. 2A, 2B, and 2C respectively illustrate methodologies 200, 230, and 270 for implementing techniques disclosed herein using cloud computing data processing system 152 or similar systems.

[0078] Beginning at FIG. 2A, methodology 200 illustrates example techniques for configuring a cloud computing data processing system for single-point, interoperable management of multiple cloud services. Methodology 200 begins at 202 and proceeds to 204 where a determination to whether a task is being taken into a task repository. If the determination at 202 returns negative, methodology 200 can proceed to 212 where a determination is made as to whether a request to assign a task to cloud resources is being made.

[0079] If the determination at 202 returns positive, task parameters can be received at 206. This can involve, *e.g.*, receiving, via an interface, task input including a plurality of task parameters. The task parameters can be arranged and provided (*e.g.*, at 214-218) to cause one or more tasks to run on cloud services.

[0080] In some embodiments, at 208 the received parameters can be encoded or converted (*e.g.*, to a local format used by a cloud computing data processing system, to a standard used by one or more cloud services) prior to populating the fields of a task repository. At 210, the task parameters (or transformed values thereof) can be stored in the task repository for use with cloud services.

[0081] At 212, a determination is made as to whether a task assignment is pending (*e.g.*, a task is manually or automatically selected to run on cloud resources). If the determination at 212 returns in the negative, methodology 200 can proceed to end at 220, or

may recycle to, *e.g.*, 202 to continue awaiting tasking. If the determination at 212 returns positive, an entry in the task repository associated with a task to run is accessed to retrieve parameters required for one or more cloud services (which can be among a plurality of disparate cloud services) to run the task.

[0082] In embodiments at 214, field values can be converted or encoded to match a target service requirement. For example, common parameters can be converted to dedicated parameters for a particular dedicated solution or shared parameters for a particular shared solution. In embodiments, such encoding or converting can be performed during or after task request generation.

[0083] At 216, a task request is generated. The task request can include all parameters necessary to assign the task to one or more cloud solutions. In embodiments, the task request can include parameters for assigning the task, or a group of tasks, to two or more disparate cloud solutions.

[0084] After the task request is generated, at 218, the task can be assigned by providing the task request to a cloud solution. The task request can be routed based on various load-balancing schemes, variable solving techniques, or observation of constraints related to one or more cloud solutions (*e.g.*, timeout lengths). If the task is to be provided to a dedicated solution, the parameters (*e.g.*, the dedicated parameters) or a consolidated instruction containing the parameters can be provided to an agent of a VM being run by a dedicated solution. If the task is to be provided to a shared solution, the parameters (*e.g.*, the shared parameters) or a consolidated request containing the parameters can be provided to a gateway for the shared solution. Thereafter, at 220, methodology 200 can end.

[0085] Continuing to FIG. 2B, methodology 230 illustrates example techniques for managing task running modes in conjunction with a cloud computing data processing system. Methodology 230 starts at 232 and proceeds to 234 where parameters associated with a task

for which a running mode is being determined are accessed. The task parameters may provide a default running mode, automated or manual task execution can define a running mode, or cloud or system context can be used to select a running mode. A graphical user interface can be used to provide such input, and can include a running mode selection option associated with one or more tasks.

[0086] At 236, based on the parameters or input, a running mode is determined. Running modes include dedicated, serverless, and hybrid. A dedicated running mode includes cloud resources for which a cloud computing data processing system or its users possess a higher level of administrative control, while a serverless running mode submits tasks to a third-party managed cloud solution for execution. The hybrid mode assigns one or more tasks to both. In embodiments, the determination at 236 can be based on costs or performance. Further, the determination as to whether to assign some or all of one or more tasks to a serverless mode can depend on whether the task(s) may exceed a timeout or otherwise fail a constraint of one of the shared solutions.

[0087] If the determination at 236 determines that a dedicated mode is being used, methodology 200 proceeds to 238 where a broker of a cloud computing data processing system contacts agents of virtual machines in one or more dedicated solutions to assign the tasks at 244.

[0088] If the determination at 236 determines that a severless mode is being used, methodology 200 proceeds to 240 where a cloud computing data processing system sends one or more transmissions bearing task parameters to one or more gateways associated with one or more shared solutions to request handling of the tasks by the shared solutions at 246.

[0089] If the determination at 236 determines that a hybrid mode is being used, methodology 200 proceeds to 240 where both agents and gateways are contacted to allocate the task(s) among the dedicated and shared solutions at 244 and 246. If a hybrid mode is

being used, a “reply to” field can be provided as a task parameter to allow status information to be passed back to a common receiver regardless of cloud solution.

[0090] In embodiments, a virtual machine of a dedicated solution can be started or stopped, or resources of a shared solution can be requested or released, to scale based on the needs of a determined running mode.

[0091] Regardless of the mode selected, one or more tasks can be allocated, in whole or part, to a single solution or two or more similar or disparate solutions. Thereafter, with tasks allocated, methodology 230 ends at 248.

[0092] Continuing to FIG. 2C, methodology 270 illustrates example techniques for managing scaling and load-balancing in conjunction with a cloud computing data processing system. Methodology 270 begins at 272 and proceeds to 274 where a task is identified for assignment and parameters accessed. Thereafter, at 278, performance metrics are accessed to assess resource availability and where resources are available.

[0093] Thereafter or simultaneously, at 282, a determination is made as to whether the load balancing assignment requires use of a free agent (*e.g.*, VM not currently involved in a session). If so, a free agent is located and methodology 270 proceeds to 288 where the task is assigned to the free agent.

[0094] Regardless of whether a free agent receives the task, methodology 270 proceeds to 284 where a determination is made as to whether a particular cloud solution should scale up or allocate additional resources. This is because, if no free agent is used, additional resources still might be required, and if a free agent is used, another free agent may need to be created to observe a minimum free agent count in a cloud computing data processing system. (In an alternative embodiment, the task identified can be completing or terminating, and an auto-scaling determination can relate to scaling down a VM or releasing resources.) A scale-up instruction can be based on resource utilization or other metrics in

conjunction with service levels, task urgency, and various scaling thresholds. The values analyzed prior to issuing the instruction can be current or realtime, historical, or projected.

[0095] If the determination at 284 returns negative, no scaling is required, and the task is assigned to an existing virtual machine if not yet assigned. If the determination at 284 returns positive, methodology 270 proceeds to 286 where additional resources are allocated. Thereafter, if the task is not yet assigned, it can be assigned at 288 to the newly-allocated resources made available through auto-scaling. Thereafter, at 290, methodology 270 can end.

[0096] In alternative or complementary embodiments, a scale-down determination or instruction can be performed in parallel or as a separate routine. A determination as to whether to scale down resources, based on performance metrics or other information, can be made at 294. If the determination returns negative, methodology 270 ends at 290. If the determination at 294 returns positive, methodology 270 proceeds to 296 where resources are scaled down. In embodiments, a deconfliction check can be completed prior to scaling down to ensure tasks are complete prior to terminating a VM or releasing resources. Further, a free agent check can be conducted to ensure scaling down does not reduce free agents below a target number. A scale-down instruction can be based on resource utilization or other metrics in conjunction with service levels, task urgency, and various scaling thresholds. The values analyzed prior to issuing the instruction can be current or realtime, historical, or projected. Thereafter, methodology 270 ends at 290.

[0097] In alternative or complementary embodiments of FIG. 2C, load balancing and scaling can be performed in the absence of a new task. For example, tasks can be dynamically reallocated on the fly to continuously load balance. In another example, auto-scaling can occur based on the completion of tasks rather than the assignment of tasks. Other variants to methodology 270, as well as methodologies 200 and 230, will be apparent on review of this disclosure.

[0098] Fig. 2D illustrates an example methodology 2000 for deploying a virtual machine image to create an instance of the virtual machine image on one or more of a disparate plurality of cloud solutions. Methodology 2000 begins at 2002 and proceeds to 2004 where a scale-up instruction can be identified. The scale-up instruction can specify particular parameters (*e.g.*, resources requested, tasks to be handled, user, service level) for a virtual machine being instantiated in response to the scale-up instruction. Based on various load balancing or auto-scaling rules, the scale-up request can be designated to one or more cloud solutions, which may be shared or dedicated solutions. The designated solution will dictate a particular format (*e.g.*, the deployed format) of image to be provided to accordingly instantiate a VM, as each solution may require a particular or proprietary image format to properly instantiate a particular virtual machine image.

[0099] At 2006, a virtual machine image satisfying the scale-up instruction can be identified. In embodiments, the virtual machine image is stored in an image database. The image database can store images in native formats or in a stored image format, which can standardize the images according to a common format. In embodiments, 2004 and 2006 may be a single step, or may change partially in order, such that an image is identified before a target cloud solution to instantiate the image is identified.

[00100] Based on the target solution(s) and identified virtual machine image, at 2008, a determination is made as to whether the virtual machine image available matches a format which can be instantiated on the target solution(s). If the determination at 2008 returns negative, methodology 2000 proceeds to 2010 where stored image is converted from the stored image format to a deployed image format suitable for deploying to the target solution(s). Thereafter, or if the determination at 2008 returns positive, methodology 2000 advances to 2012.

[00101] At 2012, a determination is made as to whether the target solution(s) are dedicated solutions or shared solutions. This determination is not mutually exclusive, and in embodiments, both paths can be utilized simultaneously or sequentially for a single virtual machine image. If the determination at 2012 indicates that a dedicated solution is a target for instantiating the virtual machine image, the virtual image in the deployment format, or parameters related thereto, are provided to an agent of the dedicated solution at 2016. If the determination at 2012 indicates that a shared solution is a target for instantiating the virtual machine image, the virtual image in the deployment format, or other parameters related thereto, is provided to a gateway associated with the shared solution at 2014.

[00102] At 2018, the image is deployed, and a new virtual machine is instantiated (or an existing virtual machine is modified) to deploy the image. Thereafter, at 2020, methodology 2000 ends.

[00103] In alternative or complementary embodiments, a stored image format can conform to image formatting used by a target solution, and can be deployed directly to the target solution without conversion. In alternative or complementary embodiments, methodology 2000 or similar methodologies can include an image import aspect whereby a virtual machine image is generated or received and stored in an image database. In further embodiments a generated or received virtual machine image can be converted from a native format to a stored format for storage in the stored format (*e.g.*, independently or in conjunction with the native format virtual machine image). In alternative or complementary embodiments, multiple virtual machine images can be deployed to one or more cloud solutions in one or more formats.

[00104] Fig. 3 is a block diagram of network device 300 that may be connected to or comprise a component of network 100. For example, network device 300 may implement one or more portions of method 200 for placement of network components of application

102. Network device 300 may comprise hardware or a combination of hardware and software. The functionality to facilitate telecommunications via a telecommunications network may reside in one or combination of network devices 300. Network device 300 depicted in Fig. 3 may represent or perform functionality of an appropriate network device 300, or combination of network devices 300, such as, for example, a component or various components of a network, a processor, a server, a gateway, a node, or the like, or any appropriate combination thereof. It is emphasized that the block diagram depicted in Fig. 3 is example and not intended to imply a limitation to a specific implementation or configuration. Thus, network device 300 may be implemented in a single device or multiple devices (*e.g.*, single server or multiple servers, single controller or multiple controllers). Multiple network entities may be distributed or centrally located. Multiple network entities may communicate wirelessly, via hard wire, or any appropriate combination thereof.

[00105] Network device 300 may comprise a processor 302 and a memory 304 coupled to processor 302. Memory 304 may contain executable instructions that, when executed by processor 302, cause processor 302 to effectuate operations associated with managing cloud services. As evident from the description herein, network device 300 is not to be construed as software per se.

[00106] In addition to processor 302 and memory 304, network device 300 may include an input/output system 306. Processor 302, memory 304, and input/output system 306 may be coupled together (coupling not shown in Fig. 3) to allow communications there between. Each portion of network device 300 may comprise circuitry for performing functions associated with each respective portion. Thus, each portion may comprise hardware, or a combination of hardware and software. Accordingly, each portion of network device 300 is not to be construed as software per se. Input/output system 306 may be capable of receiving or providing information from or to a communications device or other network

entities configured for telecommunications. For example input/output system 306 may include a wireless communications (*e.g.*, 3G/4G/GPS) card. Input/output system 306 may be capable of receiving or sending video information, audio information, control information, image information, data, or any combination thereof. Input/output system 306 may be capable of transferring information with network device 300. In various configurations, input/output system 306 may receive or provide information via any appropriate means, such as, for example, optical means (*e.g.*, infrared), electromagnetic means (*e.g.*, RF, Wi-Fi, Bluetooth®, ZigBee®), acoustic means (*e.g.*, speaker, microphone, ultrasonic receiver, ultrasonic transmitter), or a combination thereof. In an example configuration, input/output system 306 may comprise a Wi-Fi finder, a two-way GPS chipset or equivalent, or the like, or a combination thereof.

[00107] Input/output system 306 of network device 300 also may contain a communication connection 308 that allows network device 300 to communicate with other devices, network entities, or the like. Communication connection 308 may comprise communication media. Communication media typically embody computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. By way of example, and not limitation, communication media may include wired media such as a wired network or direct-wired connection, or wireless media such as acoustic, RF, infrared, or other wireless media. The term computer-readable media as used herein includes both storage media and communication media. Input/output system 306 also may include an input device 310 such as keyboard, mouse, pen, voice input device, or touch input device. Input/output system 306 may also include an output device 312, such as a display, speakers, or a printer.

[00108] Processor 302 may be capable of performing functions associated with telecommunications, such as functions for managing cloud services, as described herein. For example, processor 302 may be capable of, in conjunction with any other portion of network device 300, managing multiple disparate cloud services by assigning tasks, managing running modes, conducting scaling and balancing, *et cetera*, as described herein.

[00109] Memory 304 of network device 300 may comprise a storage medium having a concrete, tangible, physical structure. As is known, a signal does not have a concrete, tangible, physical structure. Memory 304, as well as any computer-readable storage medium described herein, is not to be construed as a signal. Memory 304, as well as any computer-readable storage medium described herein, is not to be construed as a transient signal. Memory 304, as well as any computer-readable storage medium described herein, is not to be construed as a propagating signal. Memory 304, as well as any computer-readable storage medium described herein, is to be construed as an article of manufacture.

[00110] Memory 304 may store any information utilized in conjunction with telecommunications. Depending upon the exact configuration or type of processor, memory 304 may include a volatile storage 314 (such as some types of RAM), a nonvolatile storage 316 (such as ROM, flash memory), or a combination thereof. Memory 304 may include additional storage (*e.g.*, a removable storage 318 or a non-removable storage 320) including, for example, tape, flash memory, smart cards, CD-ROM, DVD, or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, USB-compatible memory, or any other medium that can be used to store information and that can be accessed by network device 300. Memory 304 may comprise executable instructions that, when executed by processor 302, cause processor 302 to effectuate operations to manage multiple cloud services.

[00111] Fig 4A illustrates a graphical user interface assisting with task arrangement for a cloud computing data processing system. A menu (left) generally shows task definition controls; a thread drawing canvas (gridded area) allows the drawing of threads linking tasks or creating processing threads; a thread running control (upper right) allows threads to be started, stopped, *et cetera*; and a task properties definition control (bottom center) allows for task properties to be defined for the thread.

[00112] FIG. 4B illustrates a graphical user interface showing status and resources utilization as described herein. The graphical user interface of FIG. 4B can show multiple tasks, multiple VMs, or multiple similar or dissimilar cloud solutions in terms of aggregated resources and aggregated resource utilization, unlike other task interfaces which show only utilization in, *e.g.*, a single VM.

[00113] The methods and systems associated with managing cloud services as described herein also may be practiced via communications embodied in the form of program code that is transmitted over some transmission medium, such as over electrical wiring or cabling, through fiber optics, or via any other form of transmission, wherein, when the program code is received and loaded into and executed by a machine, such as an EPROM, a gate array, a programmable logic device (PLD), a client computer, or the like, the machine becomes a device for implementing content delivery as described herein. When implemented on a general-purpose processor, the program code combines with the processor to provide a unique device that operates to invoke the functionality of a streaming system.

[00114] While cloud service management systems and methods have been described in connection with the various examples of the various figures, it is to be understood that other similar implementations may be used or modifications and additions may be made to the described examples of a system or method without deviating therefrom. For example, one skilled in the art will recognize that a streaming system as described in the

instant application may apply to other environments combining both local and network elements and components. Therefore, cloud service management systems and methods as described herein should not be limited to any single example, but rather should be construed in breadth and scope in accordance with the appended claims and other disclosed embodiments.

CLAIMS

What is claimed is:

1. A cloud service management system, comprising:
an engine configured to:
receive performance metrics from two or more disparate cloud services,
wherein the two or more disparate cloud services include a dedicated solution and a shared solution, and
select one or more target resources, among the two or more disparate cloud services, to run tasks based on the performance metrics, a multiservice load balancing scheme, and task parameters;
the engine further configured to:
provide a scale-up instruction when resource utilization of the tasks, based on the performance metrics, is above a first utilization threshold, and
provide a scale-down instruction when resource utilization of the tasks, based on the performance metrics, is below a second utilization threshold and a deconfliction check is completed; and
wherein the engine is further configured, in response to the scale-up instruction, to:
command the dedicated solution to instantiate a new virtual machine to receive at least a portion of the tasks, or
provide a shared resource request message to the shared solution in preparation for handling at least a portion of the tasks; and
wherein the engine is further configured to, in response to the scale-down instruction:
command the dedicated solution to terminate an existing virtual machine, or
provide a shared resource release message to the shared solution.

2. The cloud service management system of claim 1, wherein the performance metrics are retrieved from the shared solution using a routine proprietary to the shared solution.
3. The cloud service management system of claim 1, wherein the performance metrics are retrieved from an agent associated with a virtual machine of the dedicated solution.
4. The cloud service management system of claim 1, wherein the performance metrics include projected performance and wherein the resource utilization includes projected utilization.
5. The cloud service management system of claim 1, wherein the scale-up instruction and the scale-down instruction are further based on a minimum number of free agents.
6. The cloud service management system of claim 1,
wherein the performance metrics include a dedicated solution virtual machine cost and a shared solution resource cost,
wherein the scale-up instruction commands the dedicated solution to instantiate a new virtual machine when the dedicated solution virtual machine cost is less than the shared solution resource cost,
wherein the scale-up instruction provides the shared resource request message to the shared solution when the shared solution resource cost is less than the dedicated solution virtual machine cost,

wherein the scale-down instruction commands the dedicated solution to terminate the existing virtual machine when the dedicated solution virtual machine cost is greater than the shared solution resource cost, and

wherein the scale-down instruction provides the shared resource release message when the shared solution resource cost is greater than the dedicated solution virtual machine cost.

7. The cloud service management system of claim 1, wherein the load balancing scheme is based on CPU settings associated with the tasks.

8. The cloud service management system of claim 7,
wherein a first CPU setting provides that the target resource employs a custom loading algorithm.

9. The cloud service management system of claim 8, wherein the custom loading algorithm includes a class, wherein the class includes a routine, wherein the routine includes aspects to define a configuration set, receive task objects associated with the tasks and a list indicating resources for allocation or deallocation.

10. The cloud service management system of claim 7,
wherein a second CPU setting provides that the target resource has CPU availability greater than or equal to the second CPU setting.

11. The cloud service management system of claim 7,

wherein a third CPU setting provides that the target resource is dedicated to the one of the tasks having the third CPU setting, and wherein the target resource remains dedicated until the one of the tasks is complete.

12. A method, comprising:

receiving performance metrics from two or more disparate cloud services, wherein the two or more disparate cloud services include a dedicated solution and a shared solution, and

selecting one or more target resources, among the two or more disparate cloud services, to run tasks based on the performance metrics, a multiservice load balancing scheme, and task parameters;

providing a scale-up instruction when resource utilization of the tasks, based on the performance metrics, is above a first utilization threshold, and

providing a scale-down instruction when resource utilization of the tasks, based on the performance metrics, is below a second utilization threshold and a deconfliction check is completed; and

in response to the scale-up instruction, to:

commanding the dedicated solution to instantiate a new virtual machine to receive at least a portion of the tasks, or

providing a shared resource request message to the shared solution in preparation for handling at least a portion of the tasks; and

in response to the scale-down instruction:

commanding the dedicated solution to terminate an existing virtual machine,

or

providing a shared resource release message to the shared solution.

13. The method of claim 12, wherein the load balancing scheme is based on CPU settings associated with the tasks.

14. The method of claim 12, wherein the performance metrics are retrieved from the shared solution using a routine proprietary to the shared solution, and wherein the performance metrics are retrieved from an agent associated with a virtual machine of the dedicated solution.

15. The method of claim 12, wherein the scale-up instruction and the scale-down instruction are further based on a minimum number of free agents.

16. The method of claim 12, wherein the performance metrics include projected performance and wherein the resource utilization includes projected utilization.

17. A cloud service management system including non-transitory computer readable media storing instructions that when executed by a processor perform aspects which:

receive performance metrics from two or more disparate cloud services, wherein the two or more disparate cloud services include a dedicated solution and a shared solution, and

select one or more target resources, among the two or more disparate cloud services, to run tasks based on the performance metrics, a multiservice load balancing scheme, and task parameters;

provide a scale-up instruction when resource utilization of the tasks, based on the performance metrics, is above a first utilization threshold, and

provide a scale-down instruction when resource utilization of the tasks, based on the performance metrics, is below a second utilization threshold and a deconfliction check is completed; and

in response to the scale-up instruction, to:

command the dedicated solution to instantiate a new virtual machine to receive at least a portion of the tasks, or

provide a shared resource request message to the shared solution in preparation for handling at least a portion of the tasks; and

in response to the scale-down instruction:

command the dedicated solution to terminate an existing virtual machine, or provide a shared resource release message to the shared solution.

18. The cloud service management system of claim 17, wherein the load balancing scheme is based on CPU settings associated with the tasks.

19. The cloud service management system of claim 17, wherein the performance metrics are retrieved from the shared solution using a routine proprietary to the shared solution, and wherein the performance metrics are retrieved from an agent associated with a virtual machine of the dedicated solution.

20. The cloud service management system of claim 17, wherein the scale-up instruction and the scale-down instruction are further based on a minimum number of free agents.

ABSTRACT

Systems and methods allow users to leverage multiple disparate cloud solutions, offered by disparate service providers, in a unified and cohesive manner. A system includes an engine configured to receive performance metrics from two or more disparate cloud services, select target resources among the two or more disparate cloud services to run tasks based on the performance metrics, a multiservice load balancing scheme, and task parameters. Resources can be scaled up or down in the two or more disparate cloud services based on task loads.

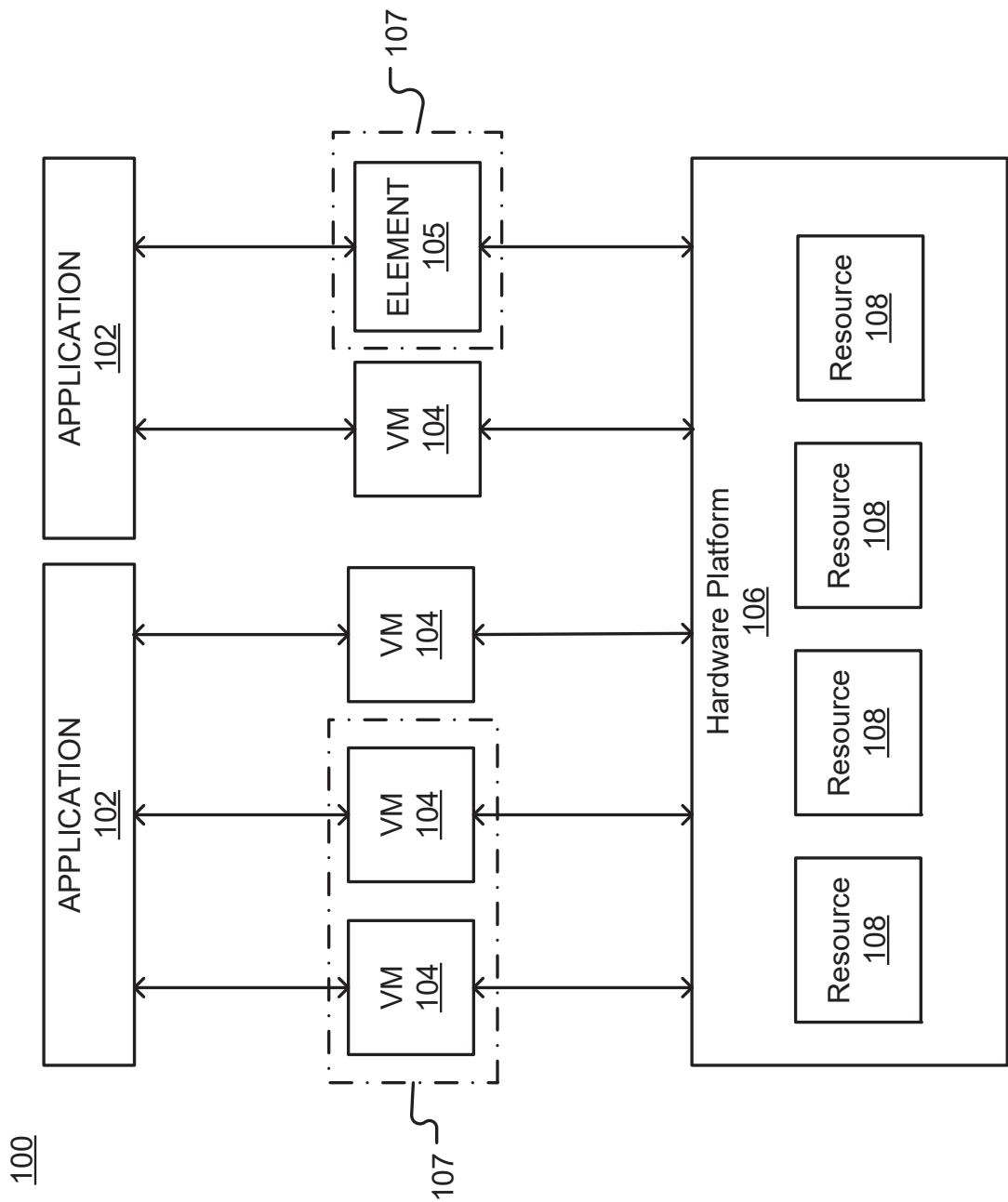


FIG. 1A

106

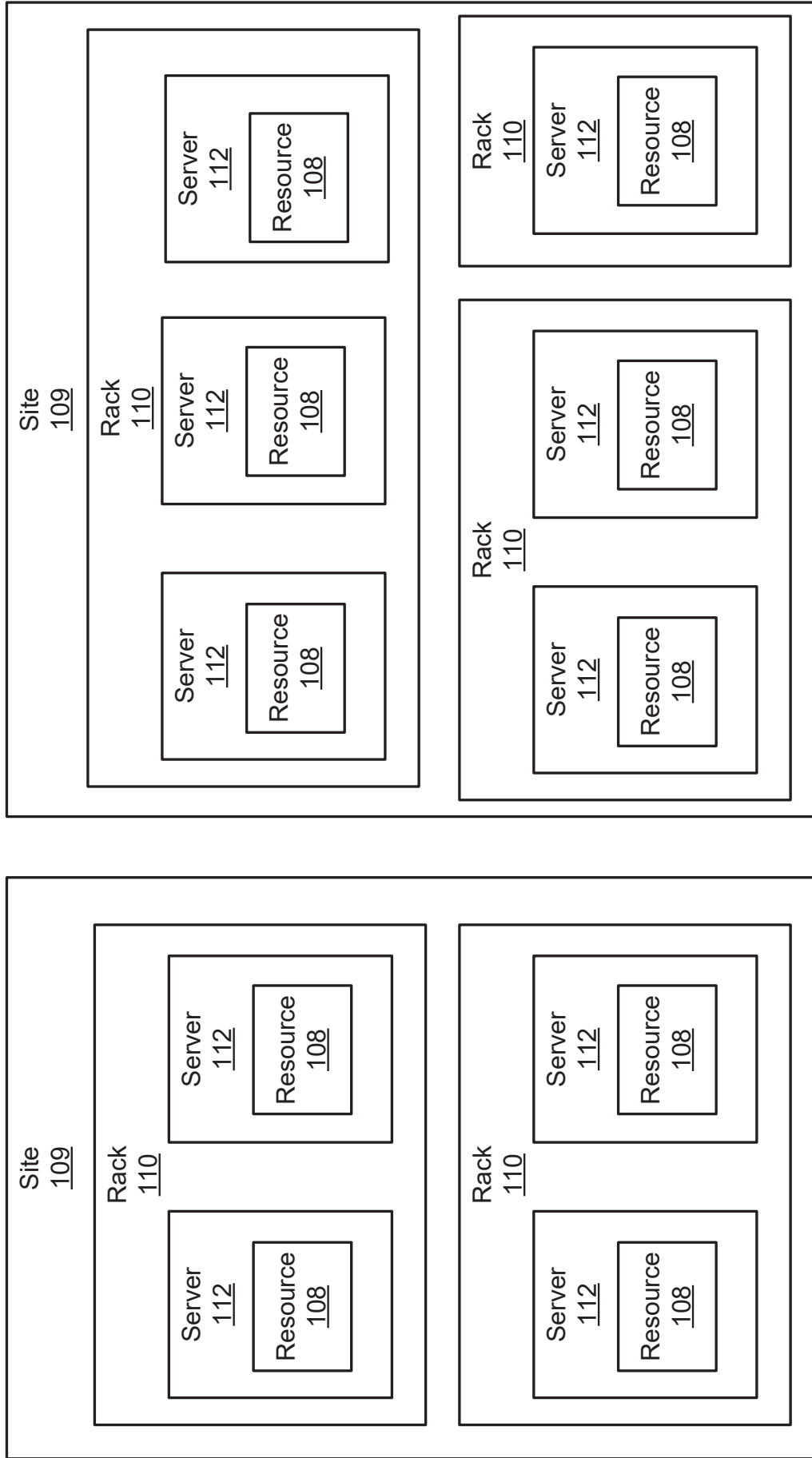
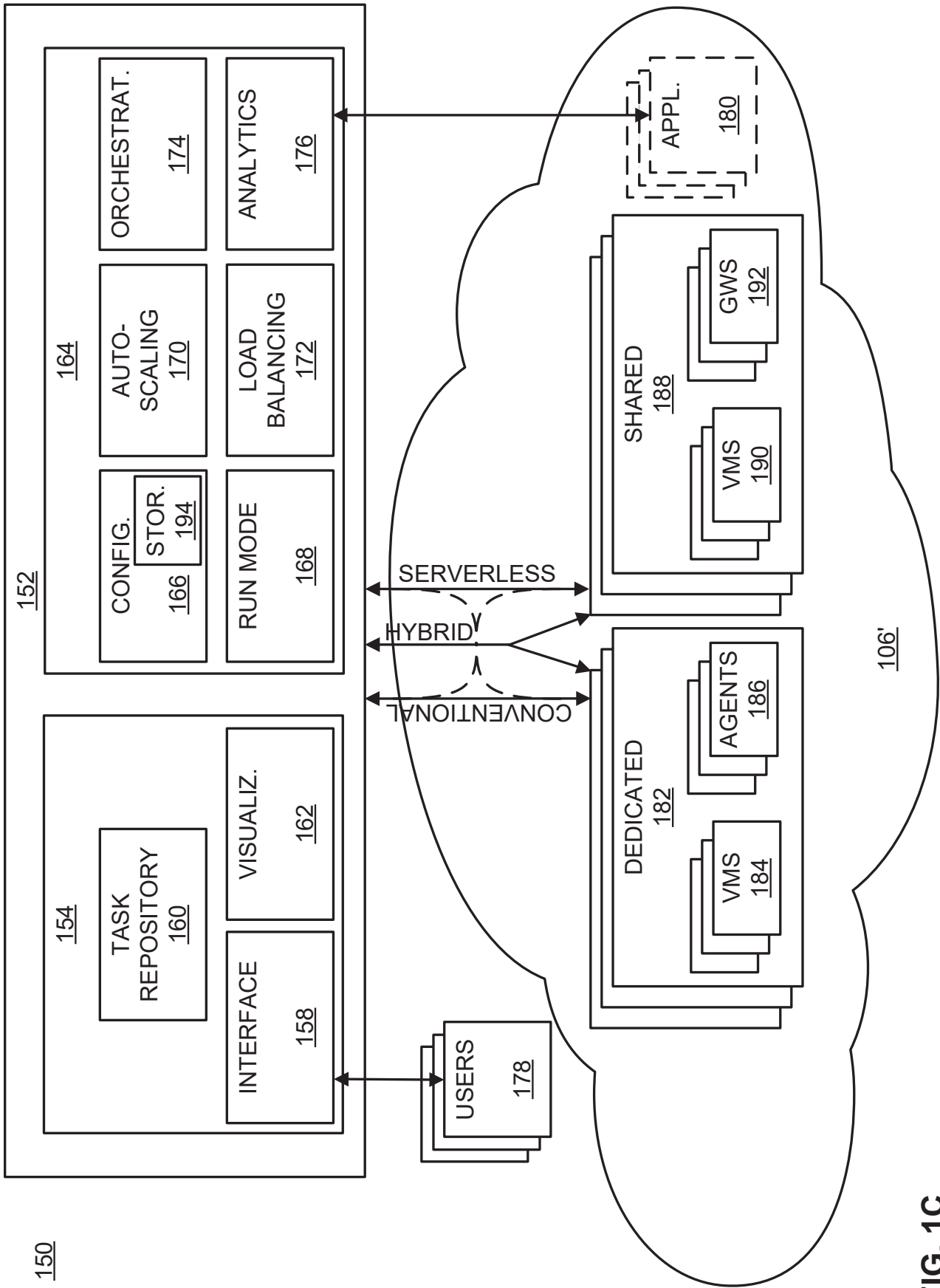
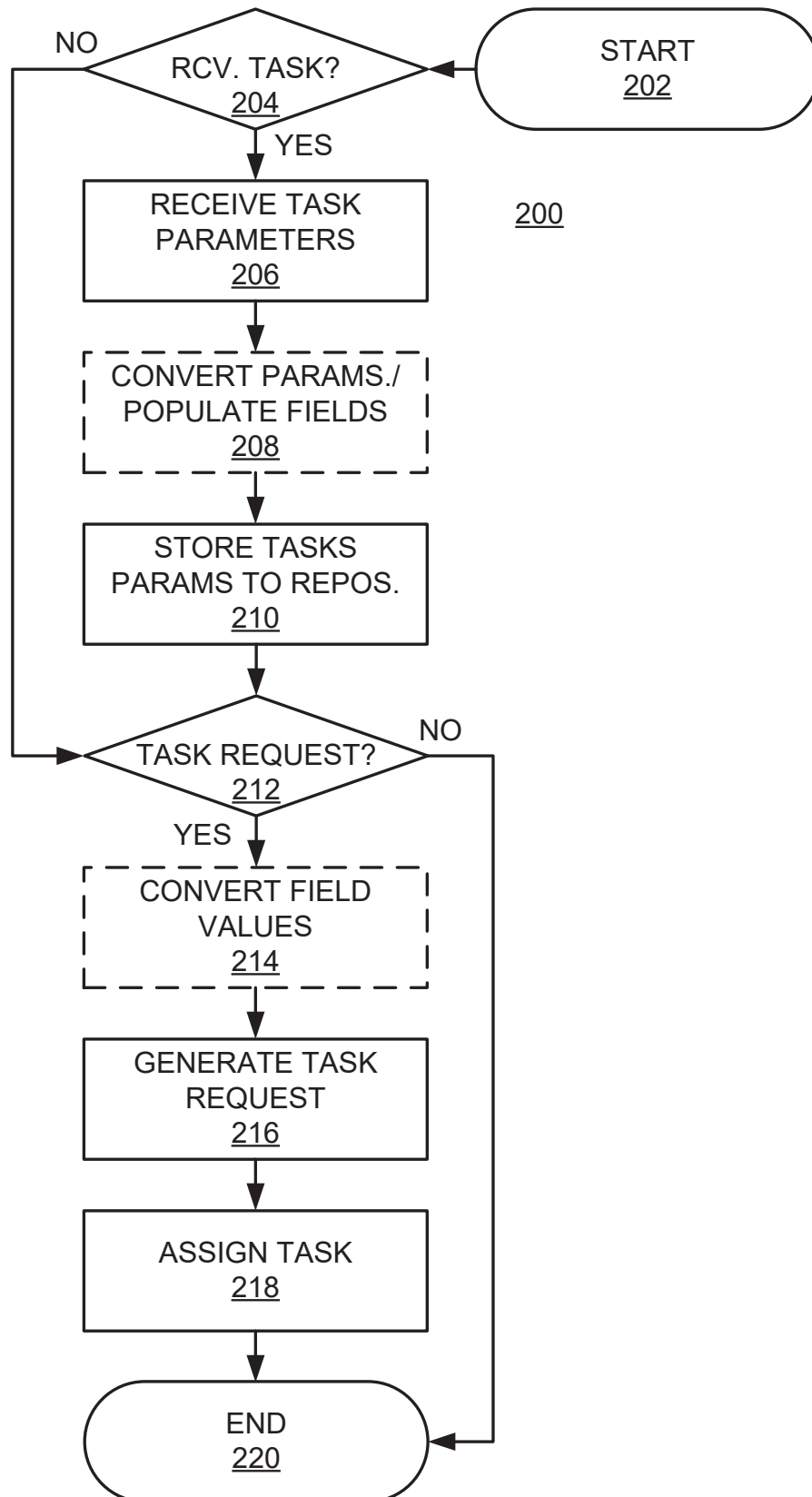
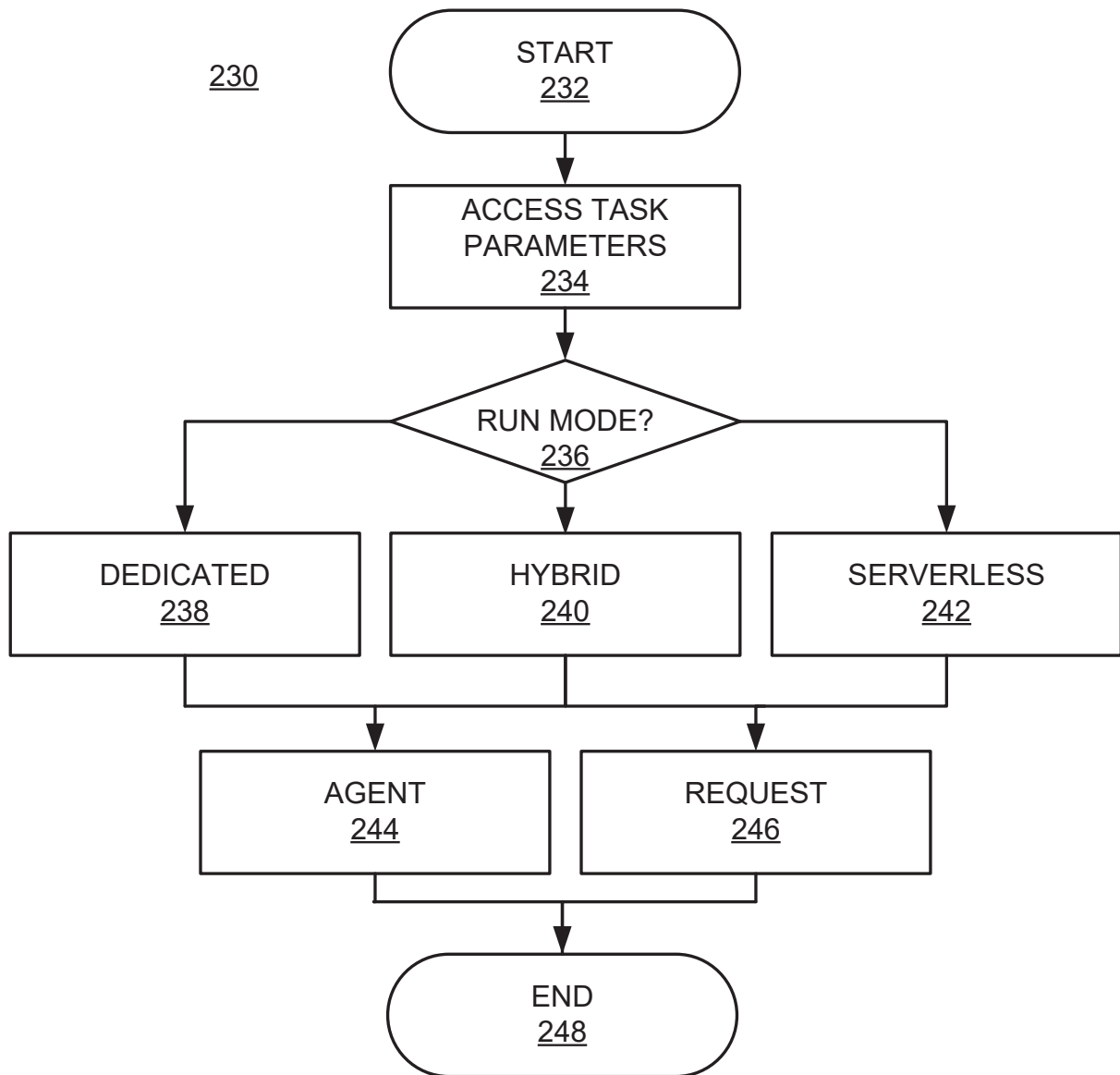


FIG. 1B



**FIG. 2A**

**FIG. 2B**

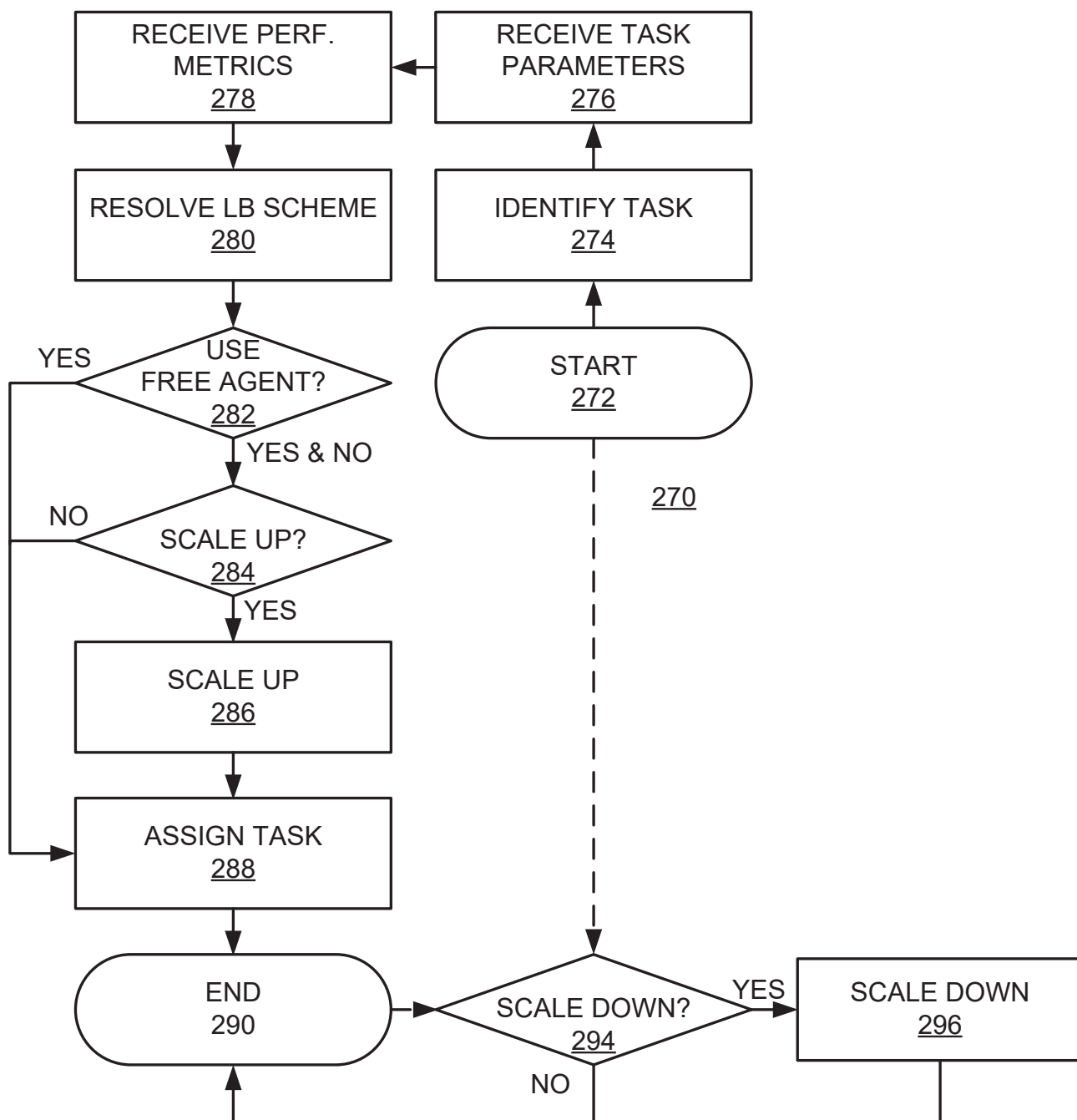
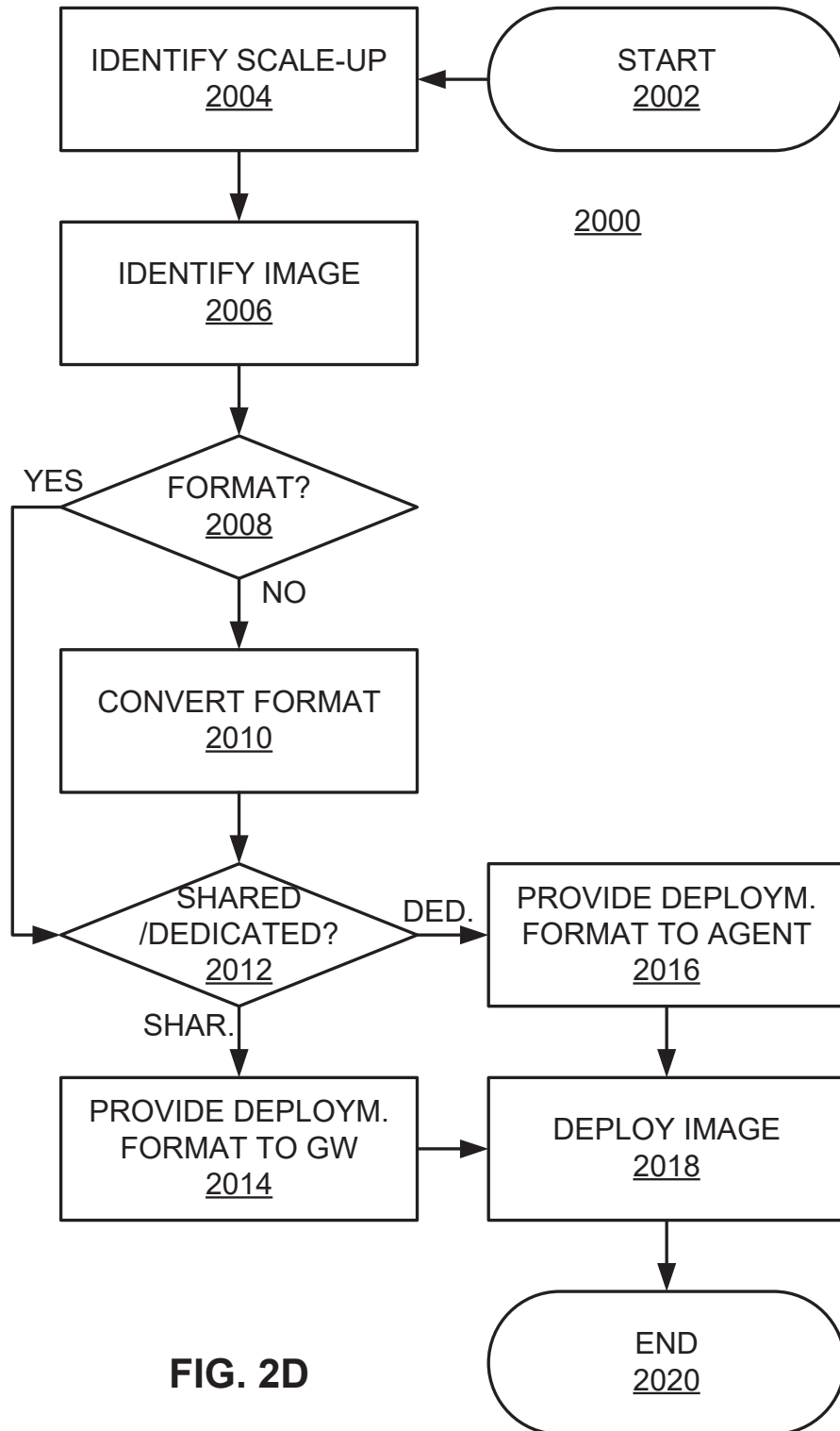


FIG. 2C



300

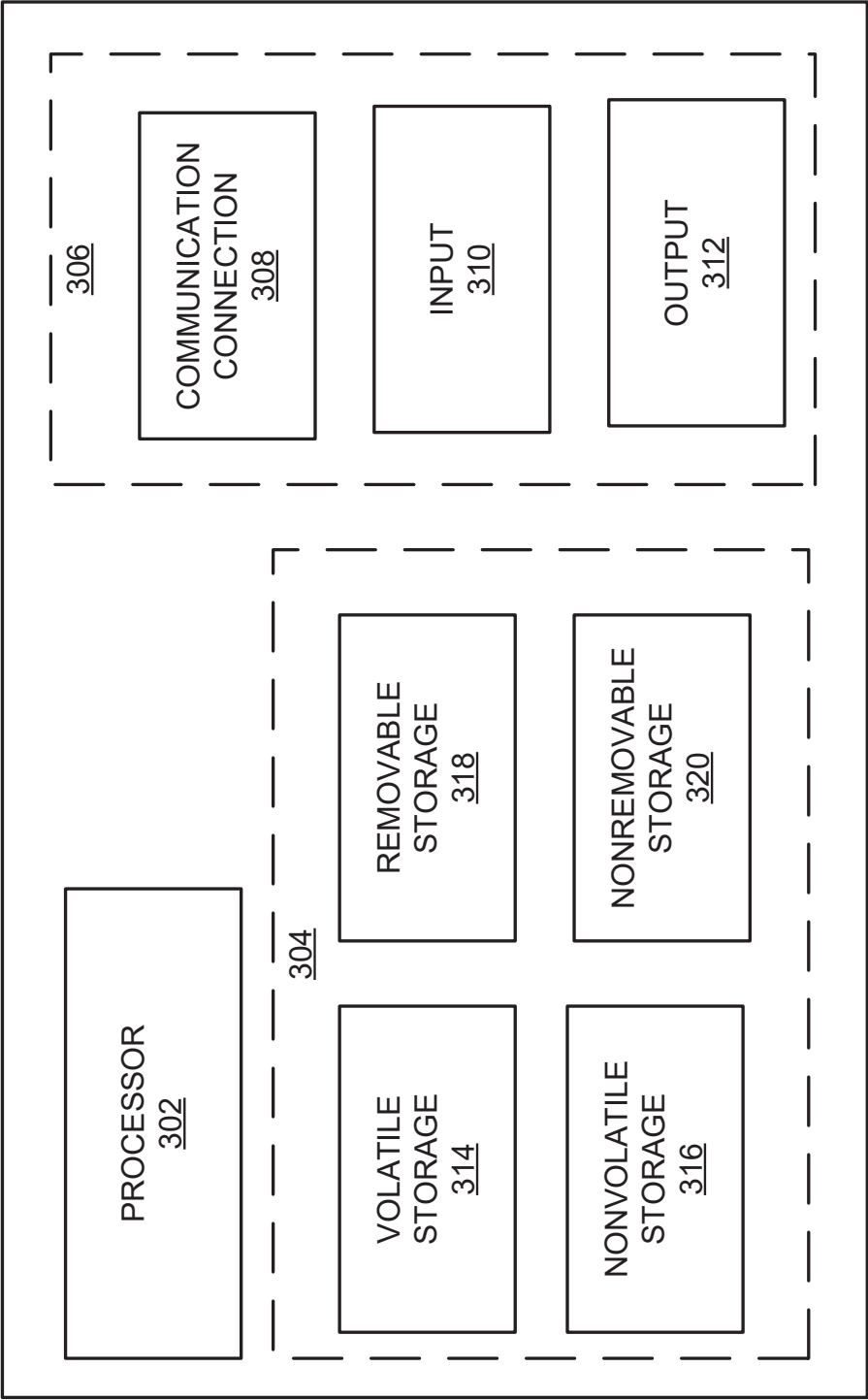


FIG. 3

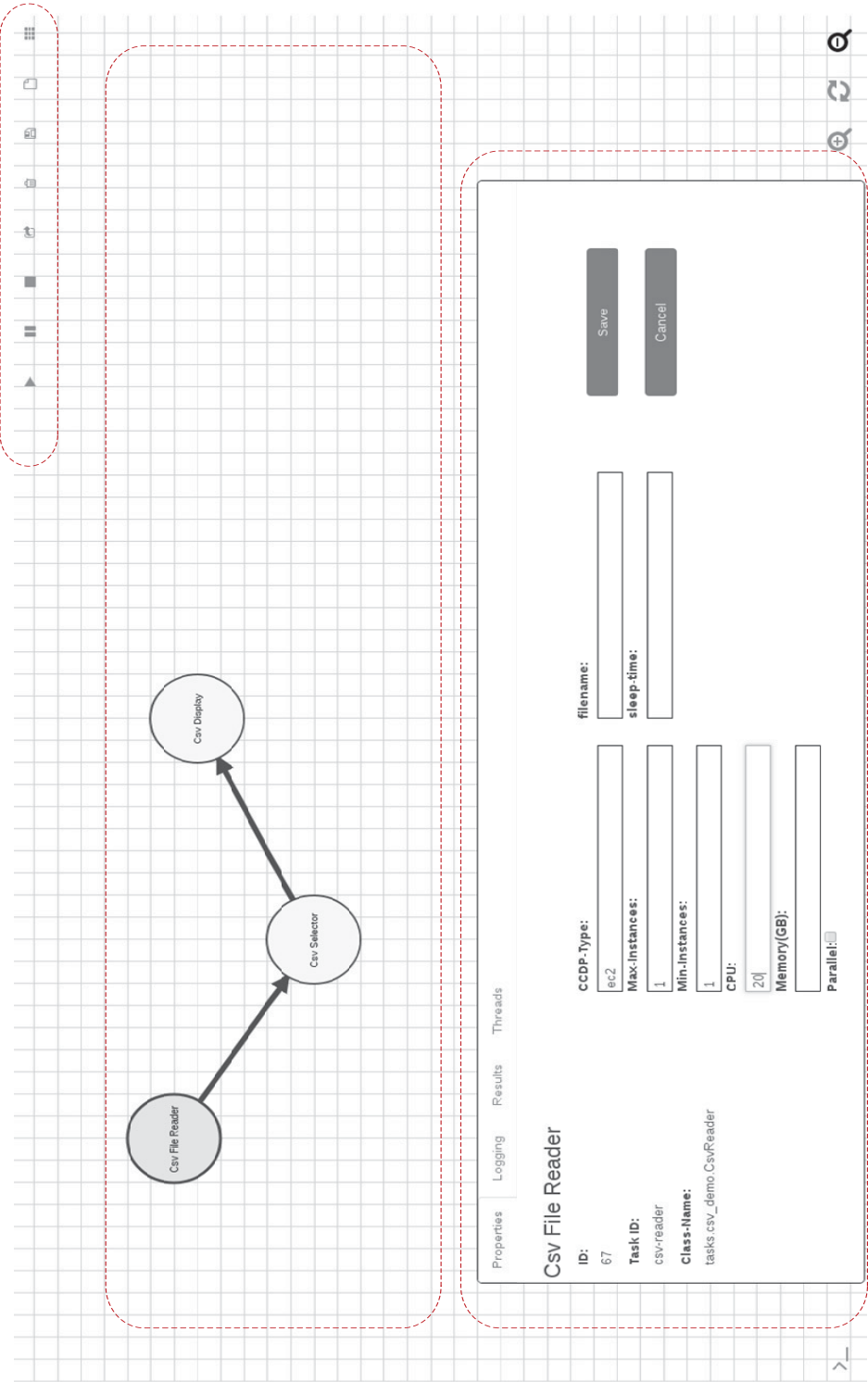


FIG. 4A

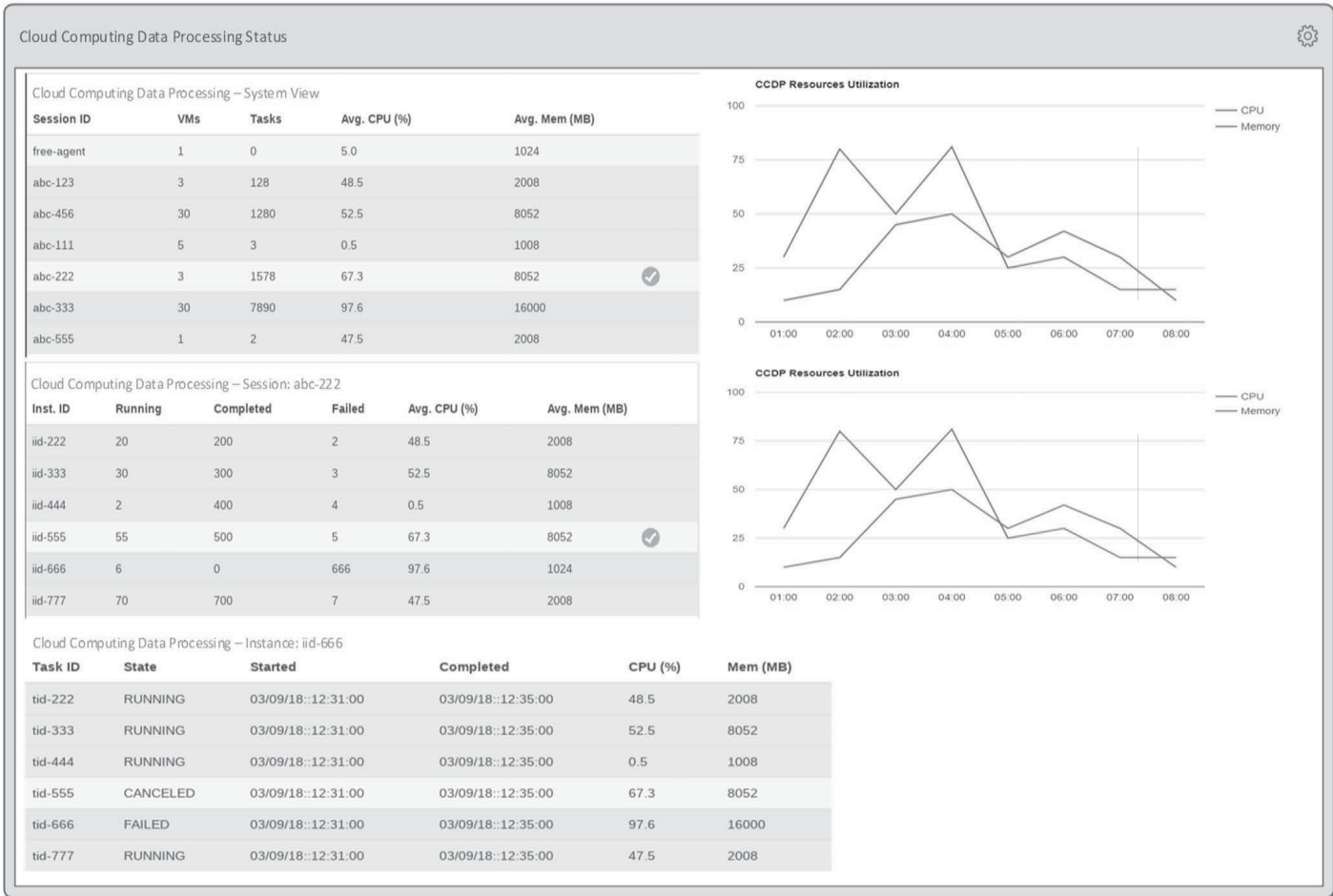


FIG. 4B

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	046850.024642
		Application Number	
Title of Invention	SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESSING		
<p>The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.</p> <p>This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.</p>			

Secrecy Order 37 CFR 5.2:

<input type="checkbox"/> Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)
--

Inventor Information:

Inventor 1					Remove	
Legal Name						
Prefix	Given Name	Middle Name	Family Name	Suffix		
	Oscar	E.	Ganteaume			
Residence Information (Select One) <input checked="" type="radio"/> US Residency <input type="radio"/> Non US Residency <input type="radio"/> Active US Military Service						
City	Falls Church	State/Province	VA	Country of Residence	US	
Mailing Address of Inventor:						
Address 1		6518 Kerns Road				
Address 2						
City	Falls Church	State/Province	VA			
Postal Code	22044	Country	US			
All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the Add button.						

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).		
<input type="checkbox"/> An Address is being provided for the correspondence information of this application.		
Customer Number	30734	
Email Address	patents@bakerlaw.com	Add Email Remove Email

Application Information:

Title of the Invention	SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESSING		
Attorney Docket Number	046850.024642	Small Entity Status Claimed	<input type="checkbox"/>
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	10	Suggested Figure for Publication (if any)	

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	046850.024642
		Application Number	
Title of Invention	SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESSING		

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	<input checked="" type="radio"/> Customer Number	<input type="radio"/> US Patent Practitioner	<input type="radio"/> Limited Recognition (37 CFR 11.9)
Customer Number	30734		

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.

When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending	Remove	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
Herewith	Continuation of	15/969839	2018-05-03

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	046850.024642
		Application Number	
Title of Invention	SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESSING		

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)ⁱ the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Application Number	Country ⁱ	Filing Date (YYYY-MM-DD)	Access Code ⁱ (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button.

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.

NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	046850.024642
		Application Number	
Title of Invention	SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESSING		

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	046850.024642
		Application Number	
Title of Invention	SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESSING		

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.			
Applicant 1			
<p>If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.</p>			
<input type="button" value="Clear"/>			
<input checked="" type="radio"/> Assignee	<input type="radio"/> Legal Representative under 35 U.S.C. 117	<input type="radio"/> Joint Inventor	
<input type="radio"/> Person to whom the inventor is obligated to assign.		<input type="radio"/> Person who shows sufficient proprietary interest	
If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:			
Name of the Deceased or Legally Incapacitated Inventor: <input type="text"/>			
If the Applicant is an Organization check here. <input checked="" type="checkbox"/>			
Organization Name	LGS Innovations LLC		
Mailing Address Information For Applicant:			
Address 1	13665 Dulles Technology Drive		
Address 2	Suite 301		
City	Herndon	State/Province	VA
Country	US	Postal Code	20171
Phone Number		Fax Number	
Email Address			
Additional Applicant Data may be generated within this form by selecting the Add button.			

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	046850.024642
		Application Number	
Title of Invention	SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESSING		

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	Given Name	Middle Name	Family Name	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1				
Address 2				
City			State/Province	
Country ⁱ			Postal Code	
Phone Number			Fax Number	
Email Address				

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.

Signature:

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Brendan E. Clark/			Date (YYYY-MM-DD)	2018-05-21
First Name	Brendan E.	Last Name	Clark	Registration Number	67,691

Additional Signature may be generated within this form by selecting the Add button.

Application Data Sheet 37 CFR 1.76		Attorney Docket Number	046850.024642
		Application Number	
Title of Invention	SYSTEMS AND METHODS FOR CLOUD COMPUTING DATA PROCESSING		

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

- 1 The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
- 2 A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
- 3 A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
- 4 A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
- 5 A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
- 6 A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
- 7 A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
- 8 A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
- 9 A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(c).

I hereby appoint:



Practitioners associated with Customer Number:

30734

OR

Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number

Name	Registration Number

As attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignments documents attached to this form in accordance with 37 CFR 3.73(c).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(c) to:



The address associated with Customer Number:

30734

OR

<input type="checkbox"/>	Firm or Individual Name			
	Address			
	City		State	Zip
	Country			
	Telephone		Email	

Assignee Name and Address: LGS Innovations LLC
13665 Dulles Technology Drive, Suite 301
Herndon, VA 20171**A copy of this form, together with a statement under 37 CFR 3.73(c) (Form PTO/AIA/96 or equivalent) is required to be Filed in each application in which this form is used. The statement under 37 CFR 3.73(c) may be completed by one of The practitioners appointed in this form, and must identify the application in which this Power of Attorney is to be filed.****SIGNATURE of Assignee of Record**

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee

Signature		Date	5/21/17
Name	Doug Many	Telephone	703 394 1533
Title	General Counsel		

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.