(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 July 2014 (17.07.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/110447 A1

(51) International Patent Classification:
G06F 9/50 (2006.01) *G06F 15/16* (2006.01)
G06F 17/00 (2006.01)

(21) International Application Number:
PCT/US2014/011150

(22) International Filing Date:
10 January 2014 (10.01.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/751,854 12 January 2013 (12.01.2013) US

(71) Applicant: **LYATISS, INC.** [US/US]; 295 N. Bernardo, Mountain View, CA 94043 (US).

(72) Inventor: **VICAT-BLANC, Pascale**; 448 Gold Mine Drive, San Francisco, CA 94131 (US).

(74) Agent: **OLYNICK, Mary, R.**; 2000 Hearst Ave, Ste 305, Berkeley, CA 94709 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: USER INTERFACE FOR VISUALIZING RESOURCE PERFORMANCE AND MANAGING RESOURCES IN CLOUD OR DISTRIBUTED SYSTEMS

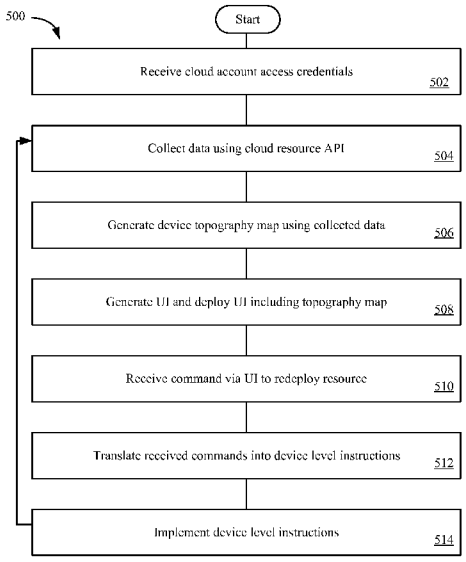


FIGURE 5

(57) Abstract: Disclosed are methods and apparatus that allow users and organizations to access on-demand and in a personalized way to the performance and optionally flow activity measures of an end to end network of virtual resources. The system is configured to generate a UI that provides a mapping and characterization of the network of virtual resources. In one embodiment, a network topology map can be generated in the UI. In another embodiment, a flow map including nodes and flows can be generated in the UI. The UI is configured to provide users with a number of actions that can be implemented to affect the virtual resources. The actions can be presented in the user via menu options or actionable objects. The user can implement high-level actions provided by the UI. Upon receipt of a high-level action, the system is configured to translate the action into a sequence of low-level device commands needed to implement the action. The system can communicate with the low-level devices in the cloud to implement the commands.

WO 2014/110447 A1

User Interface for Visualizing Resource Performance and Managing Resources in Cloud or Distributed Systems

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority of U.S. Provisional Patent Application Number 61/751,854, entitled USER INTERFACE FOR VISUALIZING RESOURCE PERFORMANCE AND MANAGING RESOURCES IN CLOUD NETWORKS, filed 12 January 2013 by Pascale VICAT-BLANC, which application is incorporated herein by reference in its entirety for all purposes.

TECHNICAL FIELD OF THE INVENTION

[0001] The invention relates to the area of managing virtual resources in a cloud or distributed environment. More particularly, the invention is related to a user interface that provides modeling and visualization of end to end network traffic and real-time resource management in the context of virtual or physical networks and in a distributed or cloud environment.

BACKGROUND

[0002] In a cloud or distributed environment, applications are distributed and deployed over virtual resources that are dynamically provisioned and mapped to a pool of physical servers that are allowed to communicate in some manner through some type of physical network. From a customer perspective, the virtual resources are typically virtual machines that execute customer applications. The machines are "virtual" in the sense that 1) the underlying physical servers on which the virtual machines are operating can change over time (migration), 2) a variable number of virtual machines are running on the same physical server, sharing the underlying processor, memory, disk and network

interface capabilities (sharing). Using the abstraction of a virtual machine, the changing nature of the physical servers is opaque to customer applications, yet, can cause the applications to experience variable and unpredictable performance.

[0003] The customer applications often include components that execute on different virtual machines that need to communicate with one another to complete a particular task. Thus, a virtual network is formed between the virtual machines where the performance of the virtual resources, including both the virtual network and the virtual machines, affects how quickly the particular task is completed within the customer application. The performance of the virtual resources is constantly changing and is difficult to characterize as the underlying physical resources are constantly changing. In addition, for a particular customer application, how the application interacts with the virtual resources affects the perceived performance of the virtual resources from the point of view of the application. This coupling between the application and the resources adds additional complexity to the performance characterization problem.

[0004] Every customer utilizing cloud or distributed network resources wants to ensure that their applications are sufficiently optimized to meet the demands of their business at all times while not wasting resources. Currently, resource management tools are very limited or non-existent. In view of the above, new methods and apparatus for application specific cloud or distributed resource management are needed.

SUMMARY

[0002] The following presents a simplified summary of the disclosure in order to provide a basic understanding of certain embodiments of the invention. This summary is not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention. Its sole purpose is to present some concepts disclosed herein in a simplified form as a prelude to the more detailed description that is presented later.

[0003] A system for allowing users and organizations to access on-demand, and in a personalized way, system and network performance and flow activity measures of an end-to-end network path is described. The system may be configured to generate a user interface (UI) that provides a mapping and characterization of the network of virtual resources. In one embodiment, a network topology map can be generated in the UI. In another embodiment, a flow map including nodes and flows can be generated in the UI. The UI is configured to provide users with a number of actions that can be implemented to affect the virtual resources. The actions can be presented in the user via menu options or actionable objects. The user can implement high-level actions provided by the UI. Upon receipt of a high-level action, the system is configured to translate the action into a sequence of low-level device commands needed to implement the action. The system can communicate with the low-level devices in the cloud to implement the commands.

[0004] In one embodiment, a method implemented in at least one electronic device including a processor and a memory is disclosed. In the processor, cloud or distributed resource access credentials for a user is received. In the processor, cloud or distributed resource data is collected using a native resource interface. In the processor, topology data is extracted from the cloud or distributed resource data, and the topology data describes virtual resources of the user in a cloud or distributed resource configuration. Based upon the topology data, a network topology map is generated in the processor. In the processor, output of a User Interface (UI) including the network topology map is controlled, and the UI includes a plurality of user selectable actions for affecting the

virtual resources. In the processor, a selection of a first action is received. The action is translated into a series of device level instructions in accordance with requirements of the cloud or distributed resource, and the processor controls communication with one or more specific virtual resources of the user to implement the device level instructions.

[0005] In a specific implementation, monitoring software is deployed to one or more virtual or physical devices associated with the user's virtual resources, and at least a portion of the cloud or distributed resource data is collected via the monitoring software. In a further aspect, based upon the cloud or distributed resource data, nodes and flows associated with the user's virtual resources are determined. a flow map that includes a plurality of actionable node or flow objects representing the flows and nodes is generated. The processor further controls output to the UI of the flow map, which includes actionable node and flow objects, which when each is selected cause additional information pertaining to the selected actionable node or flow object's corresponding node or flow to be output to the UI or cause performance of an action for managing the selected actionable node or the flow object's corresponding virtual resource. In a further embodiment, a selection of a first one of the one or more actionable node or flow objects is received and, in response to such selection, the additional information pertaining to the corresponding node or flow is displayed. In yet a further aspect, each flow represents usage of the user's cloud or distributed configuration between two or more of the nodes, and selection of a first flow object is received. In this aspect, the displayed additional information includes one or more congestion or bottleneck metrics about the first flow object's corresponding flow. In an alternative embodiment, the displayed additional information in the UI indicates performance metrics of the selected actionable node or flow object's corresponding virtual resource and the displayed additional information has a selectable mechanism for the user to change a setup of one or more performance alerts for the selected actionable node or flow object's corresponding virtual resource. In another aspect, the flow map, which is displayed in the UI, includes indications of whether each actionable node and flow object's corresponding virtual resource has a congestion or capacity level that has exceeded a predetermined threshold value.

[0006] In an alternative embodiment, the network topology map of the UI is hierarchical and includes a representation of the Internet as a root of one or more regional and/or sub-

regional networks that each includes one or more of the user's virtual resources. In a further aspect, the network topology map specifies whether each regional and/or sub-regional network and virtual resource is manageable or non-manageable via the UI. In another example, the first action specifies that a selected one of the user's virtual resources is to move from a first group of virtual resources to a second group of virtual resources. In yet another example, the first action specifies adding a network service. In another implementation, the first action specifies filtering the user's virtual resources for display in the network topology map.

[0007] In an alternative embodiment, the invention pertains to an apparatus for cloud or distributed computing resource management. The apparatus is formed from one or more electronic devices that are configured to perform one or more of the above described method operations. In another embodiment, the invention pertains to at least one computer readable storage medium having computer program instructions stored thereon that are arranged to perform one or more of the above described operations.

[0008] These and other aspects of the invention are described further below with reference to the figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Figure 1 shows a flow path abstraction, including a source and a destination, for one embodiment.

[0010] Figure 2 shows a path connecting two resources in accordance with one embodiment.

[0011] Figure 3 is diagrammatic representation of an example system that can provide flow characterization and resource management for customer cloud resources in accordance with one embodiment.

[0012] Figure 4 is a diagrammatic representation of a system providing cloud resource management in accordance with one embodiment.

[0013] Figure 5 is a flow chart illustrating a procedure for generating a user interface (UI) for managing cloud resources in accordance with a specific implementation of the present invention.

[0014] Figure 6 is an example eXecution infrastructure Description Language (VXDL) file portion.

[0015] Figure 7 shows example objects and representative symbols that can be used in a network topology map in accordance with a specific embodiment of the present invention..

[0016] Figure 8 shows a virtual network including some of the objects that are shown in Figure 7 in accordance with one embodiment of the present invention..

[0017] Figure 9 shows additional details of a manageable network object in accordance with one embodiment of the present invention.

[0018] Figure 10 shows a UI state for configuring groups of nodes in accordance with a specific embodiment.

[0019] Figure 11 shows a representation of a screen shot of the Internet at the root level, a regional network, a sub-network zone, and an end resource in accordance with a specific embodiment.

[0020] Figure 12 shows a UI state, in which a user interaction has caused information associated with a regional network to be displayed in accordance with a specific embodiment.

[0021] Figure 13 shows a UI state, in which the user has interacted with the interface to pull up a series of actions that can be performed on an end resource in accordance with a specific embodiment.

[0022] Figure 14 shows a UI state, in which the user has interacted with the interface to pull up a series of actions that can be performed on an end resource in accordance with a specific embodiment.

[0023] Figure 15 shows a UI state, in which the user has interacted with the interface to add a network service in accordance with a specific embodiment.

[0024] Figure 16 includes a screen shot with an example of one type of filtering protocol being applied in accordance with one embodiment of the present invention.

[0025] Figure 17 a is a screen shot from a user interface (UI) including a heat map in accordance with a specific embodiment.

[0026] Figure 18 is a flow chart illustrating a procedure for managing cloud resources in accordance with another embodiment.

[0027] Figure 19 illustrates a flow map in accordance with one embodiment of the present invention.

[0028] Figure 20 shows a UI state, in which the user has interacted with the UI of Figure 19 to display a usage analysis for an active end resource in accordance with a specific embodiment.

[0029] Figure 21 shows a UI state, in which the user has interacted with the UI of Figure 19 to display a usage analysis of an active flow in accordance with a specific embodiment.

[0030] Figure 22 illustrates an automatically structured flow map in accordance with one embodiment.

[0031] Figure 23 illustrates another representation of the application topology is given in the form of an actionable matrix in accordance with another embodiment of the present invention.

[0032] Figure 24 illustrates a more detailed usage analysis associated with a flow map in accordance with an alternative embodiment.

[0033] DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

[0034] In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. The present invention may be practiced without some or all of these specific details. In other instances, well known component or process operations have not been described in detail to not unnecessarily obscure the present invention. While the invention will be described in conjunction with the specific embodiments, it will be understood that it is not intended to limit the invention to the embodiments.

[0035] As described above, in the cloud or a distributed network, applications are deployed over virtual resources that are dynamically provisioned and mapped to a pool of physical servers. While this approach simplifies the server infrastructure set up, the reality of application operations is far more complex. Using this server-centric approach to cloud or distributed provisioning, enterprises that rely on the cloud or distributed computers for demanding, dynamic and mission-critical applications, expose themselves to problems including, 1) unpredictable latencies, 2) lack of visibility into resource interactions, 3) inconsistent and bad user experience, 4) disastrous effects of cascading bottlenecks and 5) wasted capacity due to over-provisioning which drives up costs.

[0036] DevOps and IT Ops teams take various approaches to resolve infrastructure problems. For example, the teams may launch multiple resources, and shut down the ones having the highest latencies or that are executing on inappropriate hardware. This approach is manual and time consuming. It leaves no opportunity for dynamic and automatic reconfiguration of the cloud or distributed networking to adapt to problems as they arise. Further, this technique involves heterogeneous management tools, many complicated scripts and manual touch points, which have errors of their own, and such errors may exacerbate rather than solve the problems.

[0037] As another approach, complicated and numerous dashboards and alarm systems can be used. These dashboards drown a user in raw data and noise and are not intuitive to use. Further, a significant portion of the raw data that is presented is not even useful to characterizing and solving the current problem of interest. Thus, sifting through these

alarms and charts and identifying the true actionable information is time consuming and error prone.

[0038] In summary, monitoring and troubleshooting applications that have increasingly varying traffic loads across multiple geographies and with complex constraints, based on a "black box network" and multiple management and operations systems, is very challenging. Applications often experience periods of bad performance and downtime while DevOps teams spend their limited time and resources with hands-on operations to continuously adapt to the changing workloads, instance failures and cloud outages. Therefore, DevOps teams need more network infrastructure automation and programmability, which can be manifested as integrated tools, to overcome these challenges.

[0039] Software Defined Networking (SDN) defines a new architecture for the "network machine" and is possibly a step in trying to address these issues. SDN decouples the Network Data plane from its Control plane to enable agile virtual networks. However, as described above and in more detail below, there are a host of issues associated with optimizing application performance in the cloud or distributed network that SDN does not directly address. Thus, the flexibility provided by SDN alone can't ensure predictable and high application performance cloud networking nor efficient cloud or distributed network operations.

[0040] As described herein, it is believed that tools for characterizing, abstracting and managing a tight coupling between application, cloud or distributed infrastructure and a virtual or physical network, SDN-based or not, can be used to address and fix the problems associated with optimizing application performance. This approach can be referred to as Application Defined Networking (ADN). ADN provides tools for adapting the network and working around problems automatically, thus maintaining business continuity and optimizing resource utilization. Although the following embodiments are described in the context of virtual networks and cloud environments, embodiments of the present invention may be applied to physical networks, distributed environments, and any other types of environments having an operating system (OS), such as Linux and the like.

[0041] ADN tools can characterize cloud operations, present characterization data in a manner that provides an intuitive understanding of current cloud performance issues and then generate an interface for presenting and implementing intelligent remedial actions for solving performance problems. As an example, using ADN tools, a user can discover and articulate the cloud network topology and the application topology associated with their particular applications. The tools can be configured to detect and locate the bottlenecks and provide options for working around these bottlenecks in real time. The ADN tools can be configured to enable two-way communication of performance and configuration information between applications and the networked infrastructure to allow applications to be adapted to the network and the network to be adapted to the applications in a cohesive manner.

[0042] With respect to the following figures, architecture for defining and characterizing network and infrastructure performance in an ADN environment is described. The architecture includes a number of abstractions that are presented for the purposes of illustration only and are not meant to be limiting as different abstractions can be utilized within the architecture. Although not described in detail herein, the characterization metrics derived from the characterization architecture can be presented in an interface that allows a user to manage and optimize their application performance and resource utilization in a cloud environment. In one embodiment, the characterization metrics can be derived from state variables associated with the TCP protocol.

[0043] In following sections, system architecture, user interface (UI) generation, a system overview and methods that can be implemented for monitoring and managing resources are described. The system architecture section, Figures 1-2, describes quantities, such as paths and flows, which can be characterized by the system. A system overview section, Figure 3 and 4, includes components of an example system architecture for implementing one or more techniques of the present invention. The methods section, Figures 5-24, includes a description of methods that include the generation of a UI. The UI can be used to provision and manage virtual resources in the cloud. This UI can also be used to provision and manage virtual networks, such as software defined networks and dynamically provisionable network services, such as virtual load balancers or virtual

routers in a cloud or a virtual link such as a VPN (Virtual Private Network) over a public long distance network.

System Architecture

[0044] In the cloud, computing and communication resources are dynamically provisioned and managed to implement an application. The cloud includes software or hardware components which process, store or transport data in a cloud. There are resources which can be dynamically provisioned and managed by cloud infrastructure users and other which cannot. A resource can be a virtual machine, a virtual load balancer, a virtual router, a virtual switch or a virtual link. A manageable resource is a resource which can be reconfigured and monitored by a cloud infrastructure user. A provisionable resource is a resource which can be dynamically provisioned and allocated to a specific cloud user for a period of time.

[0045] Figure 1 shows an abstraction of a flow path, including a source and a destination, for one embodiment. Two components of the architecture described herein are a flow and a path. A flow or path can be an abstraction of resources between a source resource and a destination resource used to carry data between two points. In one embodiment, the flow or path starts at the source's socket layer 104a and ends at the destination's socket layer 104b. The illustrated flow moves in direction 112. In a specific example, a source process 102a may initiate a flow in the source socket layer 104a, which transmits through transport layer 106a and then IP layer 108a. A destination IP layer 108b receives data from such source IP layer 108a, which is then received through destination transport layer 106b and destination socket layer 104b, and finally received by a destination process 102b.

[0046] A source or a destination of a flow or a path can be any type of logical resource. As described above, a resource is a dynamically provisioned and manageable software or hardware component which has a functional role in an application. For example, the role may be to process, store or transport data in a cloud. In one embodiment, the resource can be a logical entity. For example, it can be a virtual machine, a network service or a storage space. A resource can also be a group of similar resources. In this case, the flow

between clustered resources is the aggregation of the individual flows between the clustered resources and the destination resource. This flow is also named a flow group.

[0047] The resource can be identified by its universally unique identifier (UUID). A UUID is an identifier standard used in software construction, standardized by the Open Software Foundation as part of the Distributed Computing Environment. The intent of UUIDs is to enable distributed systems to uniquely identify information without significant central coordination.

[0048] A resource can implement a transport layer, which multiplexes and demultiplexes data from different connections and communicates with the application processes via sockets. The connections are characterized by IP addresses and ports (*e.g.,* 104a, 104b). As example, the transport layers can be UDP or TCP. In TCP/IP, every accessible server (in this case, virtual machines) has one or more IP addresses and each of those IP addresses has a large range (0-65,535) of "ports" that can be used. Connections to servers can be made based on a combination of IP address plus port. Services running on the server that accept incoming requests designate what IP/port combination they are going to listen to, and only one service can listen to any combination at one time.

[0049] A flow can represent the data exchanged between a source and a destination during a period of time. As indicated Figure 1, a flow starts at the source transport layer 106a and ends at the destination transport layer 106b. As shown in Figure 1, a flow is an aggregation of micro-flows (or connections). A flow can be composed by different types of microflows (or connections) 110, referred to as "elephants" (high volume, long duration) or "mice" (small volume, short duration).

[0050] As described above, to optimize the implementation of an application in the cloud, metrics that characterize the accessible underlying cloud infrastructure are useful. Metrics which characterize the activity generated by the application on this cloud infrastructure are also useful. The flow represents the activity of the application in the underlying network path. In one embodiment, a flow can be characterized at a minimum by its latency and throughput, which are both functions of time. The latency can be defined as an average time it takes for information to go from a source to a destination and back. The relevant unit of measurement is typically the millisecond. The latency metric can be

applied to both flow or path objects. The throughput can be defined as a rate at which information can be reliably sent to a destination. Throughput can be expressed in terms of megabits per second (Mb/s) and it is applicable to characterizing the flow.

[0051] Other metrics can be used to characterize a flow are reliability and the number of connections. The number of connections is the number of connections composing a flow. The reliability metric can relate to packets lost and duplicated over time, a percentage of redundant information that have to been sent to recover these errors and congestion events (timeout) over time.

[0052] A path is the abstraction of the sequence of network software and hardware components between a source and a destination used to carry flow data between these two points. A path starts at the transport layer of the source and ends at the transport layer of the destination. Figure 2 shows a path 202 between two resources 204 and 206, *e.g.*, a source and a destination.

[0053] In the embodiments described herein, it can be desirable to characterize a path. As described above, a path is defined by its source and destination. In one embodiment, the path may be characterized by its latency and capacity. The bandwidth capacity of the path is the upper bound of the rate at which information can be sent to a destination. It may happen that a flow using a path exceeds the capacity of the path. In this case there is congestion event and flow packets can be lost. The location where this congestion occurs is referred to as a bottleneck.

[0054] Another example of a metric is congestion level. This metric can be used to evaluate the severity of the congestion of a path. The congestion level can be defined on a 0 to 10 scale. Level 0 is used for a network path that is never congested (which never drops packet because of buffer overflow) while a 10 corresponds to a path blocking or dropping almost all packets for more than 1 hour. The congestion level can be defined by the number of drops and the duration of the event. Congestion can be costly. Some studies give numbers such as $42K cost for one hour of network outage. Path congestion for one hour is considered as an outage.

[0055] The path latency can be defined as the average round trip time experienced by a packet forwarded in the path. The minimum path latency is the lower bound of the path

latency observed during a period of time. The latency may be expressed in milliseconds. The latency can be represented as a time function or by its statistics (min, max, mean, standard deviation, 90th percentile, 99th percentile).

[0056] The capacity can be considered as an upper bound on the amount of information that can be transmitted, stored or processed by an allocated resource. The capacity can be represented as a time function or by its statistics. For example, the path capacity is expressed in Mb/s. The path capacity is the sum of the available capacity and utilized capacity.

[0057] The latency and capacity of a path can vary over time and are not necessarily accessible directly. In particular embodiments, these characteristics can be estimated by active probing or inferred from transported data. The capacity can be represented as a time function or by its statistics.

[0058] As described above, flow or a path can start and end in the transport layer of a resource, where TCP is one example of a transport layer that can be utilized. TCP is a transport protocol which has several functions. One TCP function is to send and receive data from/to the application process. A second function is to control the congestion within the network (specifically on the network path used by the connections). In various embodiments, described herein in more detail as follows, both of the functions and variables associated with these functions (TCP variables) can be utilized. In one embodiment, TCP variables of connections between a source and a destination can be used to estimate the flow patterns as well as to detect congestions within a path.

[0059] One of the aspects of optimization and resource management may be related to identifying and responding to communication bottlenecks. A bottleneck is a spot of the infrastructure where the activity is perturbed and slowed down. A bottleneck is a problem in the cloud network that is preventing cloud resources from operating at their full capacity. For example, this could be a slow router creating network congestion or an underpowered computing resource that causes an application to slow down.

[0060] The capacity of a path is the sum of utilized capacity and available capacity. The utilized capacity is the consumed amount of information that can be transmitted by unit of time, stored or processed by a utilized allocated resource. For a path, the utilized capacity

is expressed in Mb/s. The utilized capacity corresponds to the flow throughput. The available capacity is the remaining amount of information that can be transmitted by unit of time, stored or processed by a utilized allocated resource. For a path, the available capacity is expressed in Mb/s. When the available capacity approaches zero, the flow can be considered bottlenecked. When TCP is utilized, the TCP (or TCP-friendly) connections of the flow will be forced to reduce their throughput and may experience congestion events. The congestion may materialize as packet drops and a decrease of the congestion window of the connections of the source.

System Overview

[0061] In this section, an overview of a system providing tools that allow a user to manage their cloud resource is described. For illustrative purposes, an example topology of customer's resources in the cloud and management of these cloud resources is first described with respect to Figure 3. Then, a system for implementing the resource management strategy described in Figure 3 is discussed with respect to Figure 4.

[0062] Referring back to the example of Figure 2, four resources A~D associated with a user's applications executing in the cloud are shown. In Figure 3, four flows (Flows 1 and 2 of 306a, Flow3 of 306c, and Flow 4 of 306d)) have been mapped to the resources in Figure 2. The resources and the associated flows may have been automatically discovered by the system. For instance, a user may have provided access credentials to the system that enable the system to discover the user's current usage of cloud resources.

[0063] With respect to Figure 3, resource topologies with more or less flows and more or less resources are possible. Further, different flow mappings between resources A, B, C and D including more or less flows is possible. In addition, the number of flows and the number of resources for a particular user can change over time. For example, at a first time the user may utilize four resources, at a second time a user may utilize three resources and at a third time a user may use six resources. From time to time, some of the flows may remain constant, new flows may be added or existing flows and/or resources may be terminated. Thus, the number of flows and their associated sources and destinations is provided for the purposes of illustration only and is not meant to be limiting.

[0064] Returning to the example in Figure 3, resource A can collect flow data for a first flow between resource A and B and a second flow between A and C as shown by 306a. Resource C can collect flow data for a third flow between C and B as shown in 306c. Resource D can collect flow data for a fourth flow between D and A as shown in 306d. Resource B may have the ability to collect flow data but, in this example, the collected data is not associated with any flows (306b). To enable the data collection measurement software may have been previously download to the resources.

[0065] The measurement software on each of the resources can acquire data and send the acquired data to a core 304 for processing. The data acquisition can be an ongoing process where the measurement software is acquiring at different times. The data acquired over time can be used to characterize resource performance over time. In one embodiment, the measurement software on each resource may acquire data in an asynchronous manner from one another. Thus, the core can be configured to perform operations that involve synchronizing the data received from each of the resources such that it can be output in a time consistent manner.

[0066] Besides processing the data acquired from the resources, the core can be configured to automatically discover the resources for a user, such as resources A, B, C and D, generate a topology of the resources, deploy instrumentation to collect flow data, determine the flows between the resources, process the acquired data to generate path and flow characterization metrics publish results and process the flows to generate a network graph of flows. In one embodiment, the results can be published via a UI 302 that provides flow maps and flow data visualization for the various discovered resources. Further, the UI can be used to perform actions which affect the resources.

[0005] With respect to Figure 3, a system configured to perform some of the core and UI functions is described. In Figure 4, for the purposes of illustration, an example configuration involving resource performance visualization and management for two different companies, company A and company B is discussed. Company A and company B utilize cloud resources 2. Company A and company B may each have a distinct set of customers that utilize the applications provided by each company. Company A and company B are typically unaware of each other's resource utilization in the cloud.

[0006] The cloud resources 2 are distributed in two different regions, region 4 and region 6. Typically, regions refer to separate geographic locations, such as resources located in the eastern United States and the western United States or resources located in United States and Europe. The resources are distributed to serve users of the applications in a particular geographic area. The allocation of resources in relation to demand in a particular area affects application performance. Thus, the assessment and visualization of the performance of cloud resources according to region can be important.

[0007] In the example of Figure 4, a first set of applications 12 associated with company A are executing on device 10 in region 4, a second set of applications 13 associated with company A are executing on device 12 in region 4 and a second instantiation of the first set of applications 12 associated with company A are executing on device 25 in region 6. Further, a first set of applications 14 associated with company B are executing on device 16 in region 4, a second set of applications 15 associated with company B are executing on device 20 in region 4, a second instantiation of the first set of applications 14 associated with company B are executing on device 22 in region 6 and a second instantiation of the second set of applications 15 associated with company B are executing on device 24 in region 6. As described above, the devices can refer to logical entities. For example, device 10 can be a single virtual machine or a cluster of virtual machines. In addition, a set of applications executing on a device can include multiple instantiations of one or more applications within the set where the number of instantiations within the set can change over time.

[0008] The different sets of applications can communicate with one another to complete a task. For example, the first set of applications 12 for company A on devices 10 and 25 may each communicate with the second set of applications 13 on device 11. As another example, the first instantiation of the first set of applications 14 associated with company B on device 16 can communicate with the first instantiation of the second set of applications 15 associated with company B on device 20 to complete a task. In addition, the second instantiation of the first set of applications 14 associated with company B on device 22 in region 6 can communicate with one or both of the first instantiation of the second set of applications 15 on device 20 in region 4 or the second instantiation of the second set of applications 15 on device 24 in region 6 to complete a task.

[0009] In one embodiment, proprietary monitoring software can be deployed. However, its deployment is optional. The proprietary monitoring software can be executed in conjunction with the applications to provide additional measurements that can be used to characterize application performance in the cloud. However, even without the deployment of the software, some useful performance measurements may be obtained using functions that are native to the cloud resource, such as functions available via a cloud resource API (Application Program Interface) or a Network monitoring API. Thus, embodiments with and without the proprietary monitoring software are possible. In the example of Figure 4, additional monitoring software 18 has been deployed for the applications executed by company B but not for the applications executed by company A.

[0010] The applications, the devices on which they execute and the communication patterns form a topology in cloud. As described in more detail as follows, the system can be configured to discover different sets of applications executing in the cloud including patterns of inter-device communication that the applications utilize, generate metrics as a function of time that characterize that resource performance including inter-device communication and abstract a topology. The performance information can be mapped to the abstracted topology. The topology and its associated information can be presented in a user interface (UI). Besides the topology, the UI can provide a number of different services for managing the discovered cloud resources in real-time. The topology is abstracted and visually formatted in the UI to present information in a manner that makes managing the cloud resources simple and intuitive. The topology is also encoded in an XML format so that the user can access in an online or offline manner. VXDL is for example a virtual network description language which can be expressed in XML.

[0011] In Figure 4, the cloud resource management 44 is configured to provide the functions described in the previous paragraph. Cloud resource management 44 communicates with the cloud resources 2 and generates user interfaces for managing the cloud resources. In this example, the cloud resource management 44 is shown generating two UI's simultaneously, a first one 46 for company A and a second one 50 for company B. The UI's can receive inputs that trigger actions by the cloud resource management 44, such as inputs from user 48 and user 52. The UI's can be presented remotely on company controlled devices.

[0012] The cloud resource management 44 can be implemented on one or more electronic devices including processors, memory and network interfaces. Some examples of the functions that can be provided by the cloud resource management 44 are as described as follows. Data collector 26 uses native cloud functions, such as a cloud resource API, to collects data for a resource topography map that can be output in a UI. It can automatically discover a company's resources in the cloud. This function doesn't require proprietary software deployed to and running on cloud devices. However, if the proprietary software is deployed, data acquired from 26 and the proprietary software can be combined in some manner and then output to a UI.

[0013] Data collector 28 receives data from proprietary monitoring software executing in the cloud. In one embodiment, the received data can be used to generate paths and flows that are output to the UI or to an API. Device topography generator 30 generates a device topography map with or without flows depending on the data collected. Different topography abstractions are possible. Thus, the device topography generator 30 can be configured to generate one or more different topography maps depending on the abstraction that is utilized. In one embodiment, the UI may allow a user to select from among group of different topography abstractions one or more maps to be presented in the UI.

[0014] The interface object generator 32 generates and formats data for presentation to user in UI. For example, in one embodiment, the interface object generator 32 may generate flow and path objects that are used in a device topology map. The recommendation generator 34 can be configured to analyze data acquired from the cloud resource and determine actions that may improve the performance of the applications executing in the cloud. The actions can be presented as recommendations in the UIs, such as 46 and 50, where the UI provides mechanisms for allowing a user, such as 48 or 52, to indicate they wish to implement the recommendation. The UI Generator 36 generates and controls a UI that can include recommendations, topography map and interface objects for each user (e.g., company A and company B).

[0015] The device command generator 38 can be configured to generate commands for actions triggered via the UI. Actions in the UI can be presented in a high-level format. For example, a user may indicate they wish to move an execution of an application from

a first virtual machine to a second virtual machine by dragging a symbol associated with the application from the first virtual machine and placing it in the second virtual machine using a cursor or some other control mechanism. In response to this action, the device command generator 38 can generate a sequence of low-level commands to implement the action on the two devices. For instance, commands can be generated by the UI that cause the first virtual machine to shut down a resource running application and cause a new instantiation of the resource with the application running to be generated on the second virtual machine. The action can also involve moving the entire virtual machine from one network to one another with less congestion.

[0016] The command implementator 40 communicates with specific devices to implement commands determined from the device command generator 38. The command implementator 40 can be configured to communicate with the affected resources and keep track of whether the action has been successfully completed or not. The state and action logging 42 can be configured to log actions that are implemented, such as actions triggered from inputs received via the UI. Further, the state and action logging 42 can be configured to saves snap shots of a topology maps showing a state of user resources at various times. For example, the snap shots can be taken before and after a user implements one or more actions via the UI. Then, the snap shots can be shown side by side in the interface to allow the user to visually assess whether the actions had their intended effect.

[0017] The various aspects, embodiments, implementations or features of the described embodiments can be used separately or in any combination. Various aspects of the described embodiments can be implemented by software, hardware or a combination of hardware and software. The computer readable medium is any data storage device that can store data which can thereafter be read by a computer system. Examples of the computer readable medium include read-only memory, random-access memory, CD-ROMs, DVDs, flash memory, memory sticks, magnetic tape, and optical data storage devices. The computer readable medium can also be distributed over network-coupled computer systems so that the computer readable code is stored and executed in a distributed fashion.

Methods for Cloud Resource Management including an Actionable UI

[0067] Next, with respect to this section a number of methods that can be implemented using the system of Figure 4 are described. The methods can involve generating and responding to input received via a UI that is generated by the system. In a first method described with respect to Figure 5, a network topology map is generated and output to a UI. A network topology map displays the arrangement of the various elements provisioned by the Cloud user (e.g., virtual links, virtual resources) in one tenant network.

[0068] In the embodiment of Figure 5, the network topology map is generated without using proprietary software deployed to the virtual machines. In one embodiment, the network topology map can be represented in VXDL as discussed with respect to Figure 6. The UI that is generated allows a user to interact with the network topology map to initiate a number of different actions. Details of a network topology map and screen shots of the actionable UI in states that allow the actions are described with respect to Figures 7-16.

[0069] Next, in a second method, a flow map is generated. A flow map displays the application graph with flows between nodes. Flows represent the usage of the network from one virtual resource to another. The flow map can be generated using the proprietary monitoring software deployed and executed in the cloud in conjunction with the user's applications. The UI which is generated includes the flow map and can be actionable. Screen shots of the UI in states that allow the actions are described with respect to Figures 12-16.

[0070] Returning to Figure 5, a flow chart of a method 500 for generating a UI for managing cloud resources is shown for one embodiment described herein. In 502, the system can receive cloud account access credentials. For example, a login name and password can be received. The cloud account access credentials allow the system to discover and/or gain access to a user's cloud resources for modification purposes.

[0071] In 504, using the credentials, the system can collect cloud resource data. For example, the system can connect to a cloud resource API and/or parse a cloud infrastructure description file. Next, a list of cloud resources can be obtained. The list can be analyzed to extract networking information for each resource, such as UUID's and IP

addresses for each resource. The list and organization of cloud resources can also be obtained via an XML-based file (for example a VXDL file can be utilized in one embodiment).

[0072] An example of a VXDL file portion 602 is shown in Figure 6. The Virtual private eXecution infrastructure Description Language (VXDL) is a language that allows the description of a Virtual Private eXecution Infrastructure (ViPXi) or a resources graph. A VIPXI is a time-limited organized aggregation of heterogeneous computing and communication resources. It describes interconnected end resources for data processing or storage, but also the network's topology, including communication equipment and timeline representation. The ViPXi concept and its associated VXDL description language brings two aspects to the infrastructure as a service paradigm (IaaS), both related to the networking aspects: (i) the joined specification of network elements and computing elements and (ii) the link-organization concept, which permits a simple and abstract description of complex structures. The VXDL language primarily enables the description of resources and networks that are virtual.

[0073] Returning to Figure 5, in 506, a network topography map can be generated using the collected data. The map can be generated by recomposing the hierarchy of networks and resources using a selected abstraction. An internal model of the system can be created in accordance with the selected abstraction. Next, an actionable representation of each discovered network object, network service object and discovered resource object can be generated. When displayed in the UI, one or more actions may be possible from each actionable representation of an object.

[0074] In 508, the UI can be generated and deployed. For example, the UI can be output on a remote device. The UI can display the hierarchy of actionable representations of the network objects, network service objects and resource objects.

[0075] In more detail (also see Figure 11), the system can display via the UI a representation of the Internet 1102 as the "root" of the network. The Internet can be coupled to regional networks (e.g., 1104). Within the regional networks (e.g., 1104), representation of sub-networks or zones (e.g., 1106) can be displayed. In addition, zone

and sub-network information associated with sub-network or zone resources can be displayed.

[0076] Network services, which are actionable, can be displayed at their place in the hierarchy of the network topology map. General and detailed information about each network service may be made available. Further, end resources (*e.g.,* 1108), which are actionable, can be displayed at their place in the hierarchy. General and detailed information may also be provided about each end resource. In one embodiment, the Internet accessibility of each end resource can be displayed, such as a fixed or dynamic public address.

[0077] In 510, the UI can receive an input that triggers an action by the system. For example, the system can receive an action that causes a resource to be redeployed from execution on a first resource to execution on a second resource. In 512, the action can be translated into a number of device level instructions. A single action may involve multiple device level instructions. In 514, the system can implement the instructions on each of the affected devices. The automation of this process by the system can reduce the work load for users and errors associated with typing in many different device level instructions.

[0078] Next, screen shots illustrating some of the details of the method discussed with respect to Figure 5 are described with respect to Figures 8 through 16. In Figure 7, some objects and representative symbols that can be used in a network topology map are shown. The Internet, which is a ubiquitous external network, can be represented as a bar with a first color (shown as gray). A non-manageable network can be represented as bar of a second color (shown as black). A manageable network can be represented as a bar as a third color (shown as white). Manageable and non-manageable links can be represented as lines of two different colors. Manageable and non-manageable devices can be represented as boxes of two different colors. Finally, an attachment can be represented as a slender line. An attachment represents an “inclusion” relationship. Figure 8 shows a virtual network including some of the objects shown in Figure 7. In particular, a manageable network zone 804 and a non-manageable network zone 802 are each shown.

[0079] In particular embodiments, it may be possible for a user to interact with the UI such that different levels of detail associated with the virtual network are shown. In Figure 9, additional details of a manageable network object or node 902 is shown. The node 902 is a user provisionable manageable network (SDN). The node 902 includes a hardware open flow switch 904, two software switches (906a and 906b) and four virtual machines (908a~908d). A UI state for configuring groups of nodes is shown in Figure 10. Via the UI, a user may be able to use a "drag and drop" feature to add or remove elements from the different groups (1002a~1002e) and can directly access network resource configuration actions, such as changing a routing table or a flow control policy.

[0080] Next, with respect to Figures 11-16, a number of UI states including a network topology map are described. In Figure 11, a screen shot of a representation of the Internet at the root level, a regional network, a sub-network zone and an end resource are shown. In Figure 12, a UI state is shown where a user interaction has caused information 1204 associated with the regional network 1202 to be displayed. In the example, the regional network 1202 is in Northern California. In Figure 13, a UI is state is shown where a user interaction has caused information 1302 associated with an end resource 1304 to be displayed. In this example, the resource 1304 is a video server.

[0081] In Figure 14, an UI state is shown where the user has interacted with the interface to pull up a series of actions 1402 that can be performed on an end resource. In this example, the user can stop, close, move or change the instance type of the video server. In addition, a user may be able to install a "collector." The collector refers to the proprietary monitoring software described with respect to Figures 5 and 6, which can acquire data. When a collector is already installed, the user can be provided with the option of uninstalling the collector or upgrading the collector. Upgrading the collect may involve installing a new collector with added features.

[0082] As described above, implementing an action can involve translating a high-level command received from the UI into a series of low-level commands. For example, after a user has confirmed a move, the system may perform one or more of the following steps: 1) the virtual resource is stopped, 2) a snapshot of the virtual resource's root volume is taken, 3) snapshots are taken of all the storage volumes attached to this instance, 4) a new

instance is started in the new selected zone using the original instance's root volume snapshot, 5) the snapshots of the original storage volumes are attached to the new instance, 6) if the original instance has reserved public IP Address, it is moved to the new instance, 7) if one or more load balancers were pointing to the original instance, then they are updated to point to the new instance, 8) If the option "Transfer Flow history to the new Instance" is selected and a flow history is available, then the original Instance can be renamed to "<Original Instance Name> (MOVED_TO_<New Zone Name>)" and the flow data history of the original Instance can be moved to the new Instance and 9) the original Instance is stopped. If an error occurs at any step during this action, then the action can be rolled back and the original Instance can be restored along with its Elastic IP Address association (if the instance had one). The backed-up snapshots of the root and other storage volumes are not deleted in one embodiment.

[0083] The method described in the previous paragraph can be cloud resource specific depending on the provider of the cloud resource. For example, the method as applied to Amazon's cloud resource can involve one or more of the following steps, 1) the instance is stopped, 2) an EBS (Elastic Backed Storage)-backed snapshot of this instance's root volume is taken, 3) snapshots are taken of all the EBS storage volumes attached to this instance, 4) a new instance is started in the new availability zone using the original instance's root volume snapshot, 5) the snapshots of the original EBS storage volumes are attached to the new instance, 6) if the original instance has an elastic IP Address, then the elastic IP Address is moved to the new instance, 7) if one or more Amazon elastic load balancers were pointing to the original instance, then they are updated to point to the new Instance, 8) if the option "Transfer monitoring to the new instance" is selected, then the original instance will be renamed to "<Original instance name> (MOVED_TO_<New Availability Zone Name>)" and the system can transfer the monitoring history of the original Instance to the new Instance and 9) the original instance is stopped. Similar to the example above, if an error occurs during one of the steps, the action can be reversed such that the original configuration before the action is restored.

[0084] As other examples, the UI can enable a user to perform the actions of creating and configuring a load balancer. In the case of a cloud-based load-balancer, a call to API's of the provider can be made to provision the load balancer. Using the UI, a user can then be

guided through the steps needed to configure the load balancer. In one case, load-balance is a VM-based software. A call to the APIs of the provider to provision a VM can be made. Then, a software image of the load balancer can be deployed. The image may include a configuration agent. Next, the agent on the load balancer image can be activated. The core system can create a configuration file that is downloaded to the agent. The agent may configure the load per the configuration file. The agent may then restart the load balance process executing on the VM.

[0085] In Figure 15, a UI state 1502 is shown where the user has interacted with the interface to add a network service. In this example, the user is adding a load balancer. In one embodiment, the user can execute various filtering commands. For example, the UI can be configured to allow a user to filter resources by type, name, security group, and function, etc. Figure 16 includes a screen shot with an example of one type of filtering protocol being applied (1602). The resources to which the filtering criteria have been applied are highlighted.

Second Method involving a UI and including Flows

[0086] Next, a second method 1800 is described for managing cloud resources is described with respect to Figure 18. This method utilizes proprietary monitoring software deployed within the cloud. A collector described above with respect to Figure 14 is one example of proprietary monitoring software. The proprietary monitoring software can be used to add additional features to the UI, such as objects that provide a status of network communications between the various resources.

[0087] In one embodiment, the UI can generate and display a flow map. The flow map displays the application graph with flows between nodes. The flows can represent the usage of the network from one virtual resource to another. A flow list generated by the UI can display the activity and corresponding health (time-series (charts) and statistics with user-defined alert thresholds) of each flow. A node list generated by the UI can display activity and corresponding health (time-series (charts) and statistics with user-defined alert thresholds) of each node. In addition, along with the flow map, the UI can generate snap shots and heat maps. A snap shot can display quantities, such as top utilized resources (hotspots & potential bottlenecks), top flows (max throughput, max activity)

and top flow latency (highest latency). The flows can be sorted according to these different parameters. A heat map can prove a representation of network performance where the individual values (latency & available capacity) of the network path matrix are represented by gradual colors. In this matrix, the row and lines corresponding to paths with activity, the flow statistics are represented.

[0088] The visual representation can be geared toward providing information that aids in managing the cloud resources. In one embodiment as shown below in Figure 17, a flow map can be generated to display congestion or bottleneck events. The flow map includes 6 nodes (*e.g.,* 1702a-1702f) and 7 flows (e.g., 1704a and 1704b). Two nodes (1702a and 1702d) and one flow (1704b) are high-lighted because of resource issues. Node 1702a has a CPU usage greater than 90%. A second node 1702d has a disk near capacity. The flow 1704b is identified as having a high latency. A resource graph 1706 associated with the latency may be displayed to provide additional insight into the latency issue.

[0089] Next details of the method and UI states associated with the method that include flow maps are described with respect to Figures 18-22. Returning to Figure 18, in 1802, the cloud account access credentials can be obtained. Then, as described above with respect to Figure 5, an initial network topology can be generated or a previously generated network topology map can be loaded by the system. In 1804, the proprietary monitoring software can be deployed. The monitoring software can be deployed automatically by the system or manually by the user. After deployment, the UI allows a user to affect the distribution of the monitoring software. For example, a user can uninstall certain instantiations of the software on different resources if they desire.

[0090] In 1806, the system can acquire data from the cloud resource API and/or the proprietary monitoring software. In 1808, the system can set default UI settings and/or receive user specified UI settings. In one embodiment, the UI settings can affect how particular objects, such as flow objects are displayed in the interface. In 1810, a device topography map including flows can be generated. In 1812, the UI objects can be generated in accordance with the UI settings. For example, in one embodiment, a flow object can be rendered in a red color if a congestion level on the flow exceeds some threshold value selected by the user.

[0091] The generation of the flow objects can include connecting to a flow engine to get a list of active flows. Then, charts and statistical information can be generated for each flow. Next, the flow map can be constructed. The flow map can include nodes that are connected to each of the flows. Each node can represent a resource. Actionable objects which represent each node can be generated. The flow map can be organized hierarchically by tiers or by functional groups. A representation of the Internet network can be anchored at a fixed place of the screen. Next, an actionable representation for each flow can be generated. In one embodiment, a graph including flows and nodes can be output to the interface.

[0092] In 1814, action recommendations can be generated. For example, if one of the flows indicates that congestion is present, the system can be configured to generate an action for alleviating the congestion. The recommended actions can be output and presented to a user via the UI. The system can be configured to implement an action after receiving confirmation from a user. In 1816, the system can control output of the UI on a remote device. The UI can be configured to display topography maps, such as described above, flow maps, a heat map as a matrix of flows and paths, UI objects, such as graphs, and recommendations including actions that can be implemented by the user.

[0093] In 1818, user selection of an action may be received via the UI. In 1820, the system can translate actions selected via the interface to device level instructions to one or more specific devices in the cloud. In 1822, the system can communicate with one or more specific devices to implement the device level instructions. In 1824, the system can log actions and states of the cloud resources, such as before and after the action.

[0094] Next, a few screen shots from a UI including an actionable flow map are described with respect to Figures 19, 20, 21, 22. In Figure 23, another representation of the application topology is given in the form of an actionable matrix. This actionable matrix shows simultaneously the flow performance and the network performance of the non-active paths. Each cell of the matrix can be actionable such an interaction with the object in the UI can cause a generation of a new UI state that provides the user with the performance detail of flows and paths (Figure 24). In Figure 24, a detailed resource usage

analysis associated with the flow map and the heat map (or actionable matrix) is discussed.

[0095] In Figure 19, a flow map including 6 nodes (e.g., 1902a-f), which are end resources, and five flows (e.g., 1904a-d) are shown. General information 1906 (such as resource designation and IP addresses) for each active end resource can be displayed. In Figure 20, a user has interacted with the UI to display a usage analysis 2002 for an active end resource (1902a). In Figure 21, a user has interacted with the UI to display a usage analysis 2102 of an active flow (1904d). From this flow map, the user can trigger an interface state that allows a direct change to the setup of usage alerts associated with the flow's metrics.

[0096] In Figure 22, an automatically structured flow map is proposed. The structure integrates the Internet node 2204 and displays the different tiers of the application in order. The node 2202a connected to the Internet and being the only node directly accessed from the Internet, while the other nodes (e.g., 2202b and 2202c) are connected indirectly through node 2202a.

[0097] In Figure 23, another representation of the application topology is proposed. This is the representation of a graph by a matrix. In this matrix, each cell is active, such that a selection of the cell redirects the user to detailed performance information. After selection, the detailed information can be displayed in a pop up window or the UI can generate another page that displays the information.

[0098] In Figure 24, a more detailed usage analysis associated with the flow map is displayed. In one embodiment, the usage analysis includes a sparkline of a metric (2406). The sparkline can be associated with a metric that characterizes a flow or a resource. Graphs representing the states of other metrics are also shown. In addition, usage analysis for all of the metrics (e.g., 2402) is shown. The system allows a user to input actionable thresholds (e.g., 2404). The actionable thresholds are represented as lines in Figure 24.

[0099] In a particular embodiment, the system can be configured to generate alerts when one or a combination of the thresholds is exceeded. The alerts can results in instantaneous color modification in the UI. Other visual indicators can be employed. For example, objects can blink or change in brightness to attract a user's attention. In another

embodiment, an auditory cue can be generated. In yet another embodiment, the system can send out a message to a user, such as an e-mail or text, which alerts the user of the event.

[00100] Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced within the scope of the appended claims. It should be noted that there are many alternative ways of implementing the processes, systems, and apparatus of the present invention. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein.

CLAIMS

What is claimed is:

1. A method in an electronic device including a processor and a memory, the method comprising:

in the processor, receiving cloud or distributed resource access credentials for a user;

in the processor, collecting cloud or distributed resource data using a native resource interface;

in the processor, extracting topology data from the cloud or distributed resource data, wherein the topology data describes virtual resources of the user in a cloud or distributed resource configuration;

based upon the topology data, generating a network topology map in the processor;

in the processor, controlling output of a User Interface (UI) including the network topology map, wherein the UI includes a plurality of user selectable actions for affecting the virtual resources;

in the processor, receiving a selection of a first action;

translating the action into a series of device level instructions in accordance with requirements of the cloud or distributed resource; and

the processor communicating with one or more specific virtual resources of the user to implement the device level instructions.

2. The method of claim 1, further comprising deploying monitoring software to one or more virtual or physical devices associated with the user's virtual resources, wherein at least a portion of the cloud or distributed resource data is collected via the monitoring software.

3. The method of claim 2, further comprising

based upon the cloud or distributed resource data, determining nodes and flows associated with the user's virtual resources;

generating a flow map that includes a plurality of actionable node or flow objects representing the flows and nodes; and

in the processor, controlling output to the UI of the flow map, which includes actionable node and flow objects, which when each is selected cause additional information pertaining to the selected actionable node or flow object's corresponding node or flow to be output to the UI or cause performance of an action for managing the selected actionable node or the flow object's corresponding virtual resource.

4. The method of claim 3, further comprising receiving a selection of a first one of the one or more actionable node or flow objects and, in response to such selection, displaying the additional information pertaining to the corresponding node or flow.

5. The method of claim 4, wherein each flow represents usage of the user's cloud or distributed configuration between two or more of the nodes, and wherein selection of a first flow object is received, and wherein the displayed additional information includes one or more congestion or bottleneck metrics about the first flow object's corresponding flow.

6. The method of claim 3, wherein the displayed additional information in the UI indicates performance metrics of the selected actionable node or flow object's corresponding virtual resource and the displayed additional information has a selectable mechanism for the user to change a setup of one or more performance alerts for the selected actionable node or flow object's corresponding virtual resource.

7. The method of claim 3, wherein the flow map, which is displayed in the UI, includes indications of whether each actionable node and flow object's corresponding virtual resource has a congestion or capacity level that has exceeded a predetermined threshold value.

8. The method of claim 1, wherein the network topology map of the UI is hierarchical and includes a representation of the Internet as a root of one or more regional

and/or sub-regional networks that each includes one or more of the user's virtual resources.

9. The method of claim 8, wherein the network topology map specifies whether each regional and/or sub-regional network and virtual resource is manageable or non-manageable via the UI.

10. The method of claim 1, wherein the first action specifies that a selected one of the user's virtual resources is to move from a first group of virtual resources to a second group of virtual resources.

11. The method of claim 1, wherein the first action specifies adding a network service.

12. The method of claim 1, wherein the first action specifies filtering the user's virtual resources for display in the network topology map.

13. An apparatus for cloud or distributed computing resource management, the apparatus formed from one or more electronic devices that are configured to perform the following operations:

receiving cloud or distributed resource access credentials for a user;

collecting cloud or distributed resource data using a native resource interface;

extracting topology data from the cloud or distributed resource data, wherein the topology data describes virtual resources of the user in a cloud or distributed resource configuration;

based upon the topology data, generating a network topology map in the processor;

controlling output of a User Interface (UI) including the network topology map, wherein the UI includes a plurality of user selectable actions for affecting the virtual resources;

receiving a selection of a first action;

translating the action into a series of device level instructions in accordance with requirements of the cloud or distributed resource; and

communicating with one or more specific virtual resources of the user to implement the device level instructions.

14. The apparatus of claim 13, further comprising deploying monitoring software to one or more virtual or physical devices associated with the user's virtual resources, wherein at least a portion of the cloud or distributed resource data is collected via the monitoring software.

15. The apparatus of claim 14, further comprising

based upon the cloud or distributed resource data, determining nodes and flows associated with the user's virtual resources;

generating a flow map that includes a plurality of actionable node or flow objects representing the flows and nodes; and

in the processor, controlling output to the UI of the flow map, which includes actionable node and flow objects, which when each is selected cause additional information pertaining to the selected actionable node or flow object's corresponding node or flow to be output to the UI or cause performance of an action for managing the selected actionable node or the flow object's corresponding virtual resource.

16. The apparatus of claim 15, further comprising receiving a selection of a first one of the one or more actionable node or flow objects and, in response to such selection, displaying the additional information pertaining to the corresponding node or flow.

17. The apparatus of claim 16, wherein each flow represents usage of the user's cloud or distributed configuration between two or more of the nodes, and wherein selection of a first flow object is received, and wherein the displayed additional information includes one or more congestion or bottleneck metrics about the first flow object's corresponding flow.

18. The apparatus of claim 15, wherein the displayed additional information in the UI indicates performance metrics of the selected actionable node or flow object's corresponding virtual resource and the displayed additional information has a selectable mechanism for the user to change a setup of one or more performance alerts for the selected actionable node or flow object's corresponding virtual resource.

19. The apparatus of claim 15, wherein the flow map, which is displayed in the UI, includes indications of whether each actionable node and flow object's corresponding virtual resource has a congestion or capacity level that has exceeded a predetermined threshold value.

20. The apparatus of claim 13, wherein the network topology map of the UI is hierarchical and includes a representation of the Internet as a root of one or more regional and/or sub-regional networks that each includes one or more of the user's virtual resources.

21. The apparatus of claim 20, wherein the network topology map specifies whether each regional and/or sub-regional network and virtual resource is manageable or non-manageable via the UI.

22. The apparatus of claim 13, wherein the first action specifies that a selected one of the user's virtual resources is to move from a first group of virtual resources to a second group of virtual resources.

23. The apparatus of claim 13, wherein the first action specifies adding a network service.

24. The apparatus of claim 13, wherein the first action specifies filtering the user's virtual resources for display in the network topology map.

25. At least one computer readable storage medium having computer program instructions stored thereon that are arranged to perform the following operations:

receiving cloud or distributed resource access credentials for a user;

collecting cloud or distributed resource data using a native resource interface;

extracting topology data from the cloud or distributed resource data, wherein the topology data describes virtual resources of the user in a cloud or distributed resource configuration;

based upon the topology data, generating a network topology map in the processor;

controlling output of a User Interface (UI) including the network topology map, wherein the UI includes a plurality of user selectable actions for affecting the virtual resources;

receiving a selection of a first action;

translating the action into a series of device level instructions in accordance with requirements of the cloud or distributed resource; and

communicating with one or more specific virtual resources of the user to implement the device level instructions.

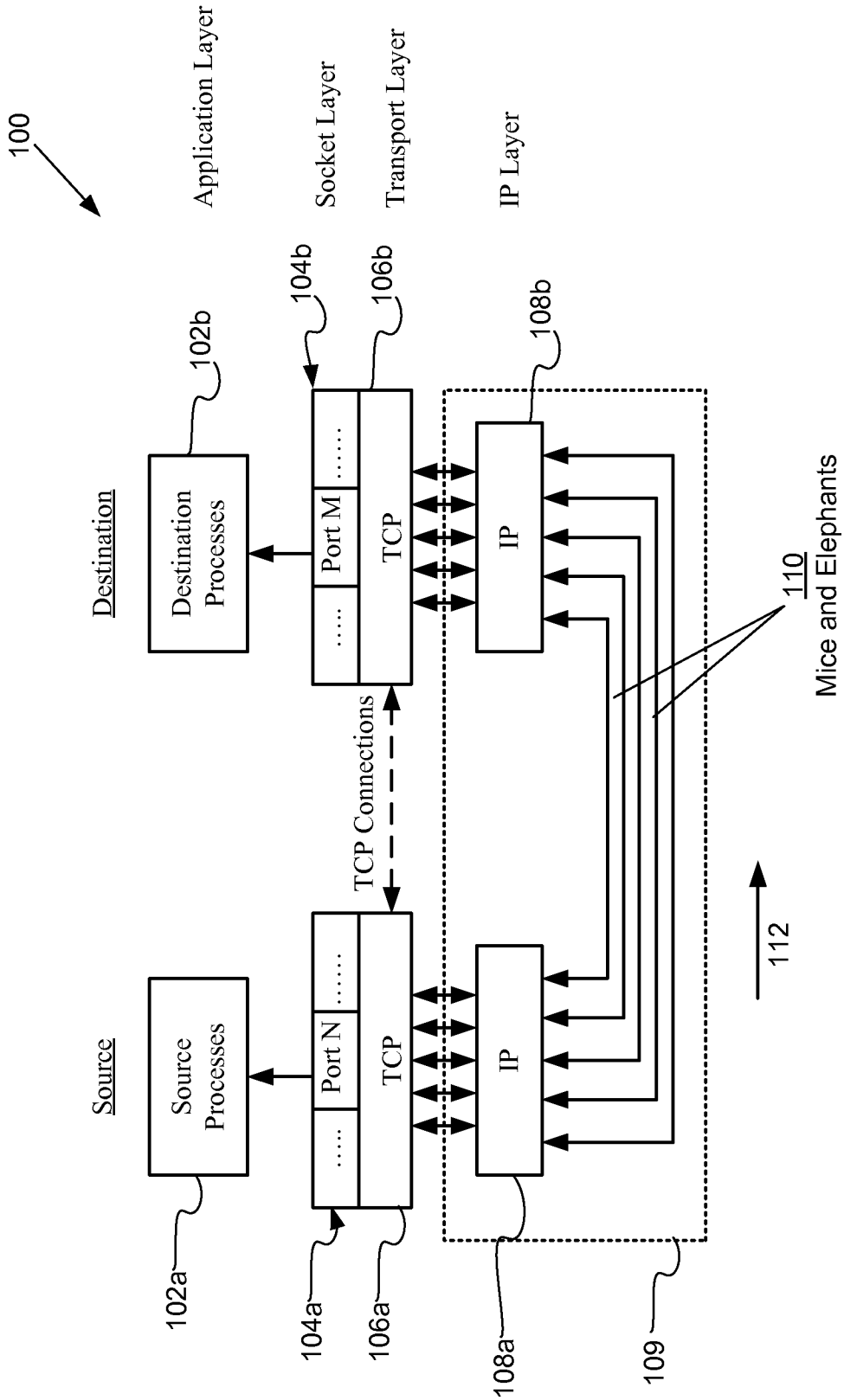

100
Application Layer
Socket Layer
Transport Layer
IP Layer
Destination
Source
Destination Processes
Source Processes
102b
102a
104b
104a
106b
106a
108b
108a
109
Port M
Port N
TCP
IP
TCP Connections
110
Mice and Elephants
112

FIGURE 1

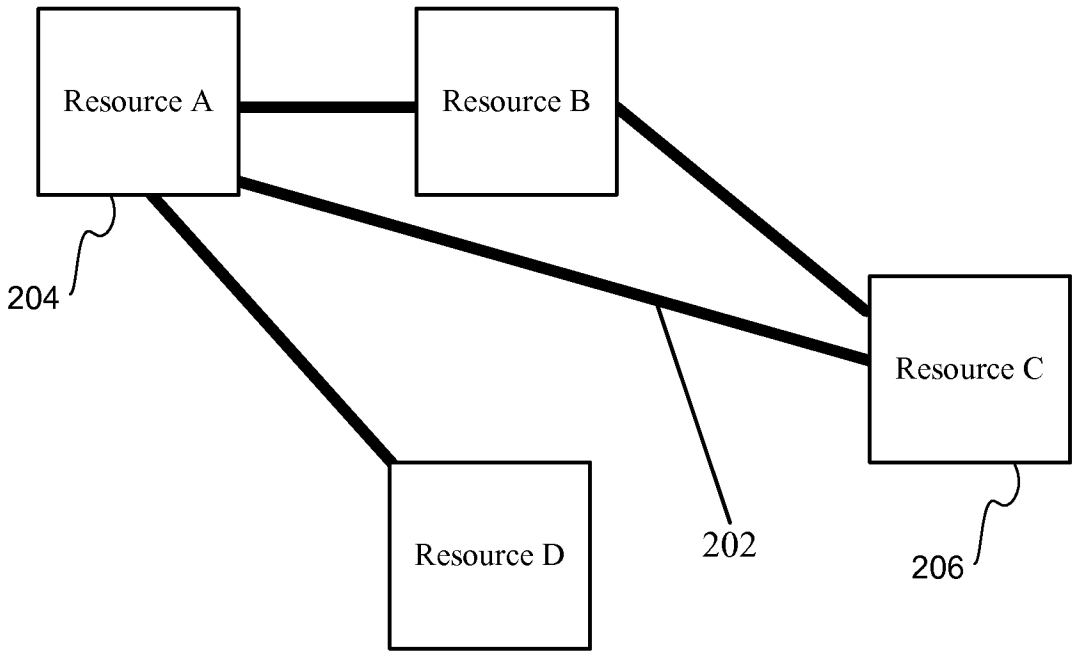


FIGURE 2

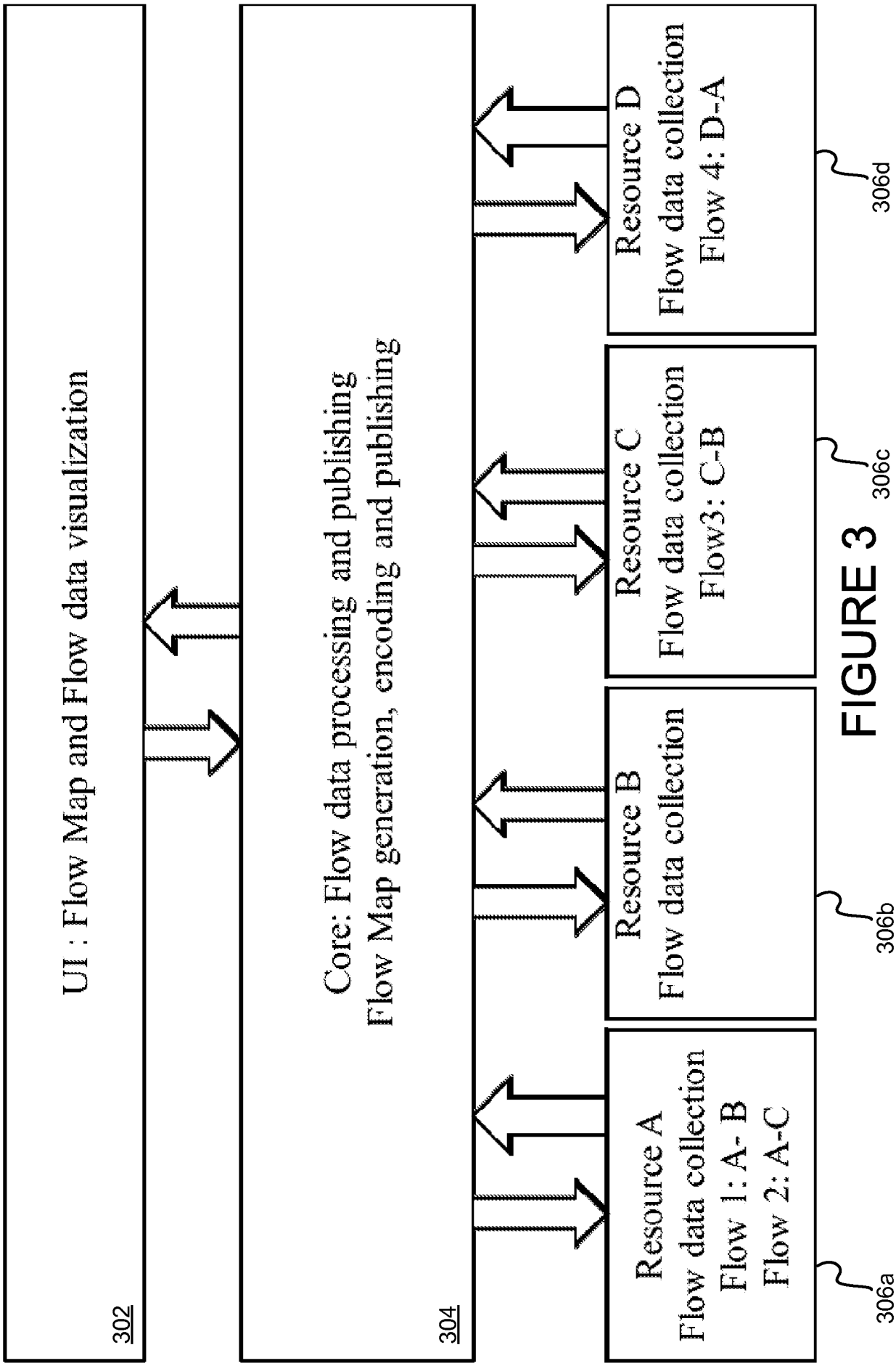

UI : Flow Map and Flow data visualization
302
Core: Flow data processing and publishing
Flow Map generation, encoding and publishing
304
Resource A
Flow data collection
Flow 1: A- B
Flow 2: A-C
306a
Resource B
Flow data collection
306b
Resource C
Flow data collection
Flow3: C-B
306c
Resource D
Flow data collection
Flow 4: D-A
306d

FIGURE 3

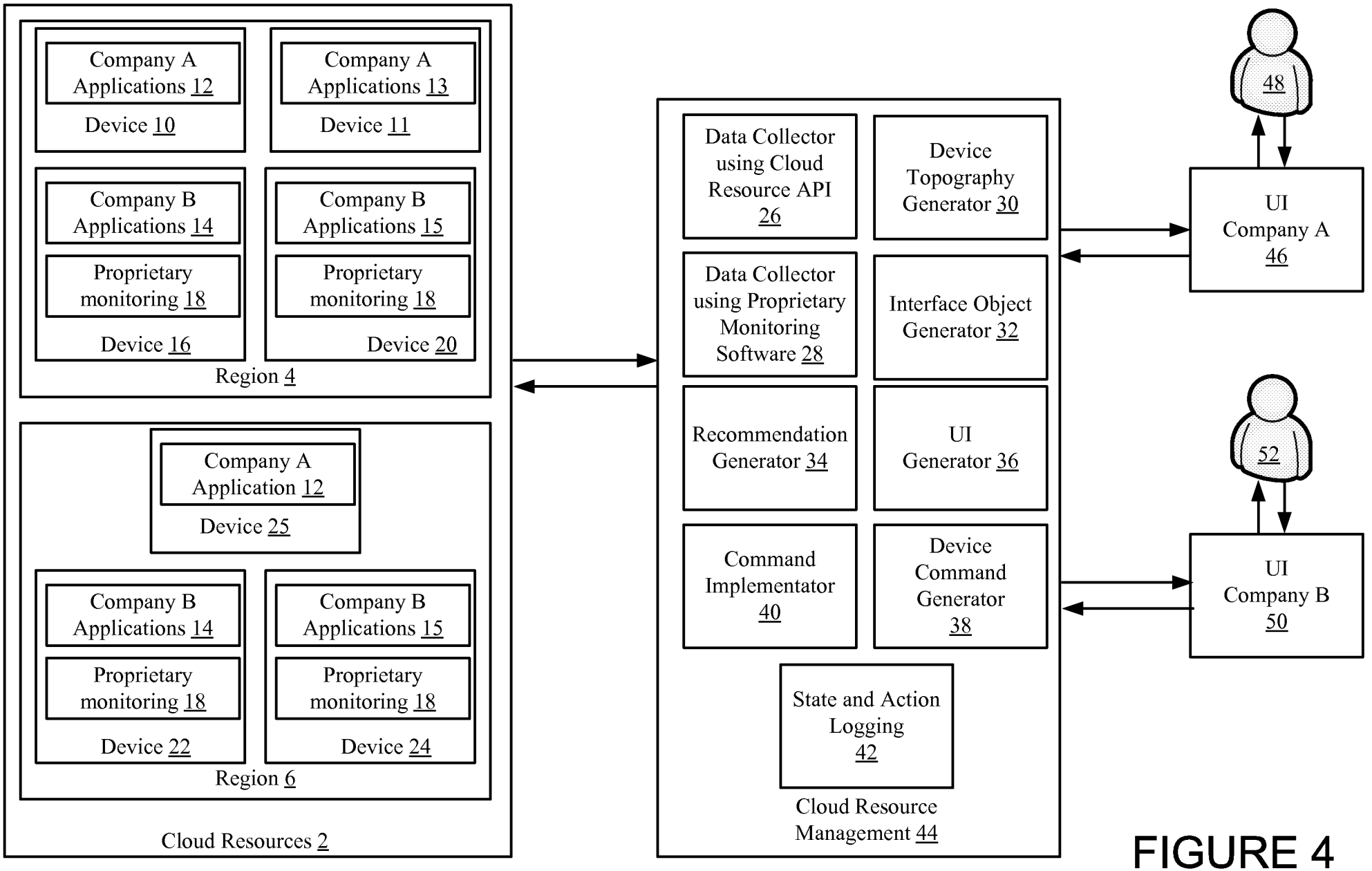

Company A Applications 12
Device 10
Company A Applications 13
Device 11
Company B Applications 14
Proprietary monitoring 18
Device 16
Company B Applications 15
Proprietary monitoring 18
Device 20
Region 4
Company A Application 12
Device 25
Company B Applications 14
Proprietary monitoring 18
Device 22
Company B Applications 15
Proprietary monitoring 18
Device 24
Region 6
Cloud Resources 2
Data Collector using Cloud Resource API 26
Device Topography Generator 30
Data Collector using Proprietary Monitoring Software 28
Interface Object Generator 32
Recommendation Generator 34
UI Generator 36
Command Implementator 40
Device Command Generator 38
State and Action Logging 42
Cloud Resource Management 44
48
UI Company A 46
52
UI Company B 50

FIGURE 4

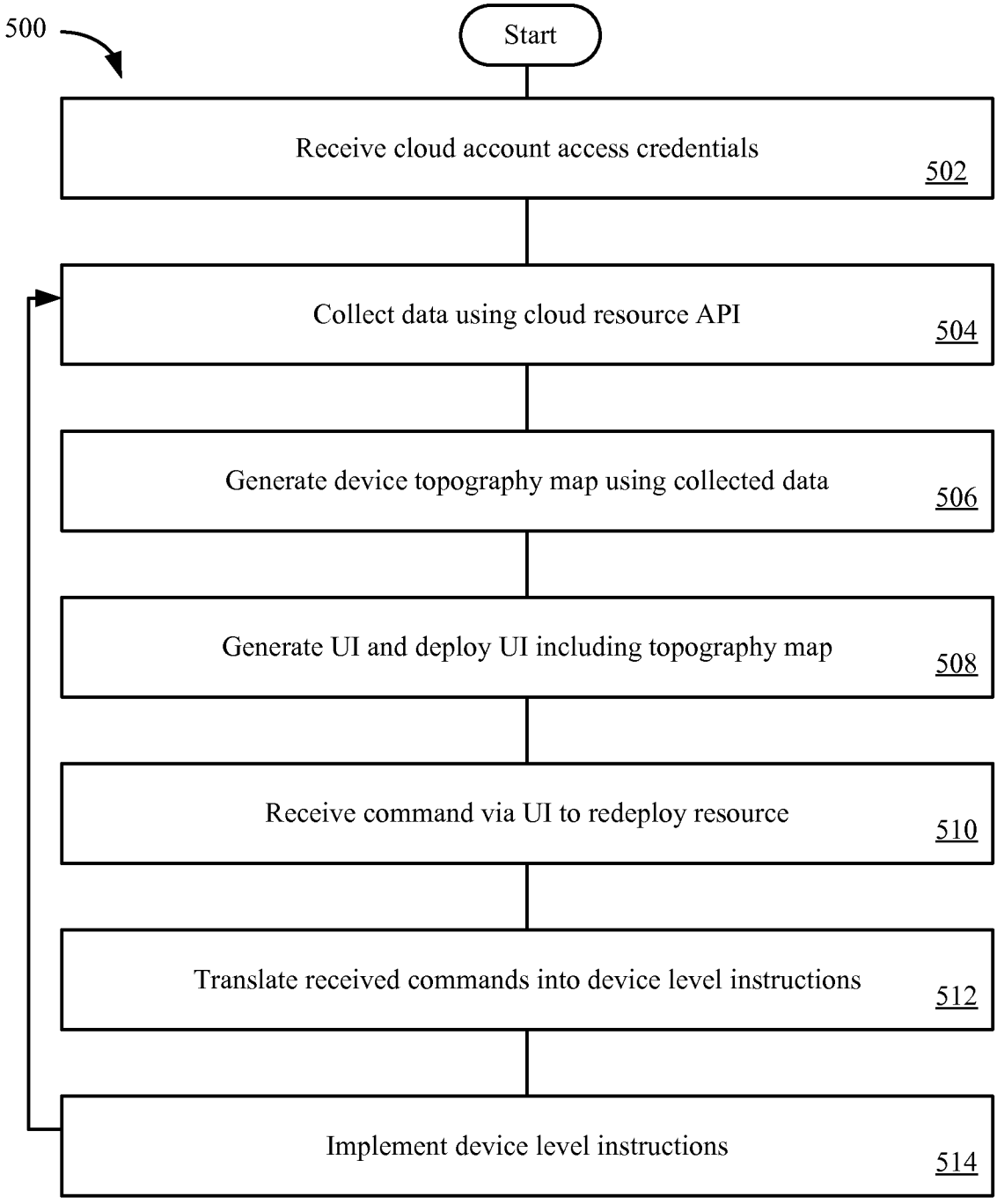
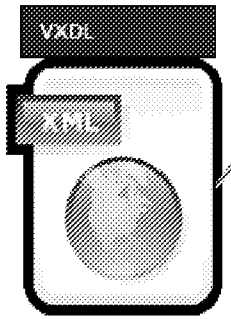
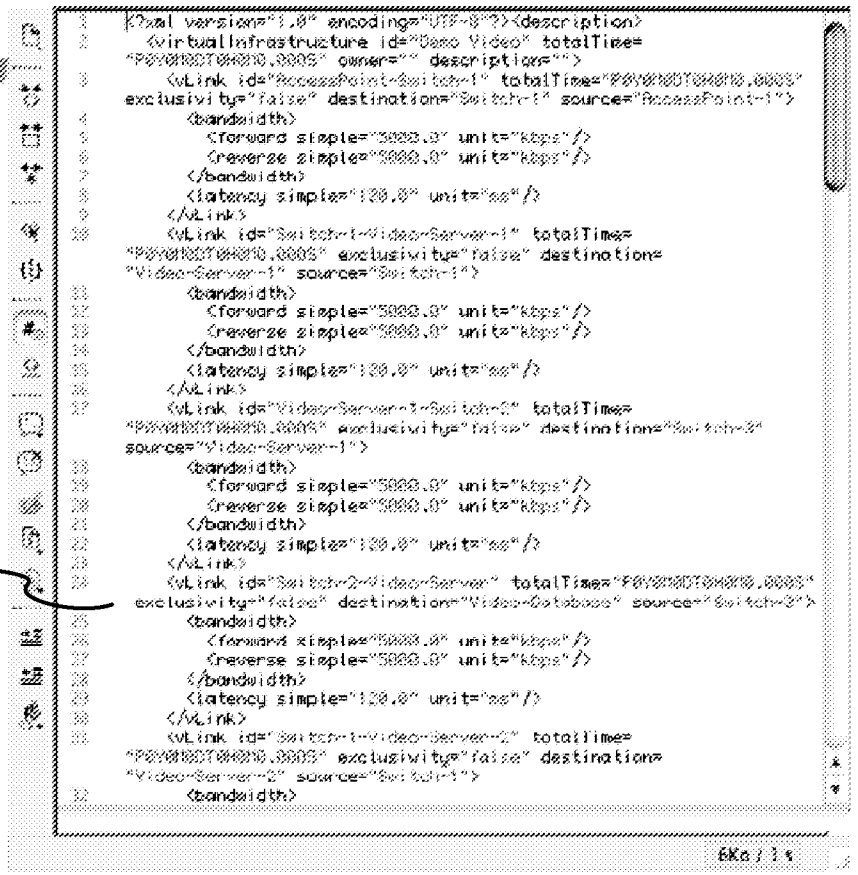

500
Start
Receive cloud account access credentials
502
Collect data using cloud resource API
504
Generate device topography map using collected data
506
Generate UI and deploy UI including topography map
508
Receive command via UI to redeploy resource
510
Translate received commands into device level instructions
512
Implement device level instructions
514

FIGURE 5

VXDL editor: virtual network language

VXDL is an XML-based language which enables descriptions of a virtual network of resources. VXDL file defines an Application Defined Network and its associated activity (flows)





- View VXDL file
- Save VXDL file
- Upload in VXDL bank
  - Access to the VXDL file repository

FIGURE 6

Object
Symbol
• Internet (external Ubiquitous network of networks)
• A non manageable network (black box principle)
• A manageable network
• A non manageable link
• A manageable link
• A manageable device
• Attachement (belong to relationship)

FIGURE 7

VXDL editor: virtual network language

VXDL is an XML-based language which enables descriptions of a virtual network of resources.
VXDL file defines an Application Defined Network and its associated activity (flows)


```
|<?xml version="1.0" encoding="UTF-8"?><description>
  <virtual Infrastructure id="Demo Video" totalTime=
"P0Y0M0DT0H0M0.000S" owner="" description="">
   <vLink id="AccessPoint-Switch-1" TotalTime="p0y0m0dt0h0m0.000s"
exclusivity="false" destination="Switch-1" source="AccessPoint-1"
        <bandwidth>
            <forward simple="5000.0" unit="kbps"/>
            <reverse simple="5000.0" unit="kbps"/>
            </bandwidth>
            <latency simple="120.0" unit="ms"/>
        </vLink>
        <vLink id="Switch-1-Video-Server-1" totalTime=
"P0Y0M0DT0H0M0.00S" exclusivity="false" destination=
"Video-Server-1" source="Switch-1">
            <bandwidth>
              <forward simple="5000.0" unit-kbps"/>
              <reverse simple="5000.0" unit="kbps"/>
            </bandwidth>
            <latency simple="120.0" unit="ms"/>
            </vLink>
            <vLink id="Video-Server-1-Switch-2" totalTime=
"P0Y0M0DT0H0M0.000S" exclusivity="false" destination="swith-3"
source="Video-Server-1">
            <bandwidth>
              <forward simple="5000.0" unit-kbps"/>
              <reverse simple="5000.0" unit="kbps"/>
            </bandwidth>
            <latency simple="120.0" unit="ms"/>
            </vLink>
        <vLink id= "Switch-2-Video-Server" totalTime="P0Y0M0DT0H0M0.000S"
        .....
```

602

- View VXDL file
- Save VXDL file
- Upload in VXDL bank
  - Access to the VXDL file repository

FIGURE 6

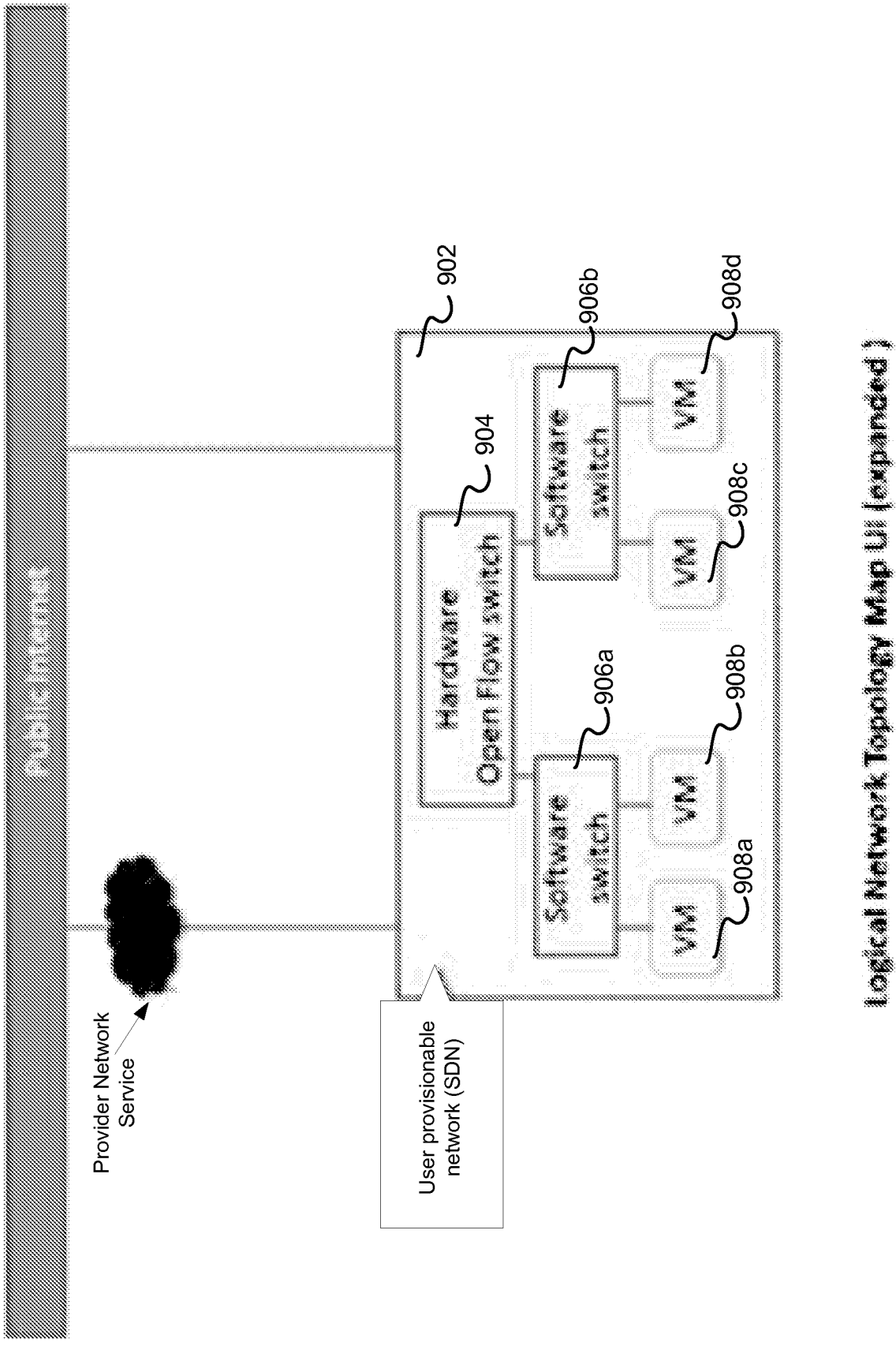

Public Internet
Provider Network Service
User provisionable network (SDN)
902
904
Hardware Open Flow switch
906a
Software switch
906b
Software switch
908a
VM
908b
VM
908c
VM
908d
VM
Logical Network Topology Map UI (expanded)

FIGURE 9

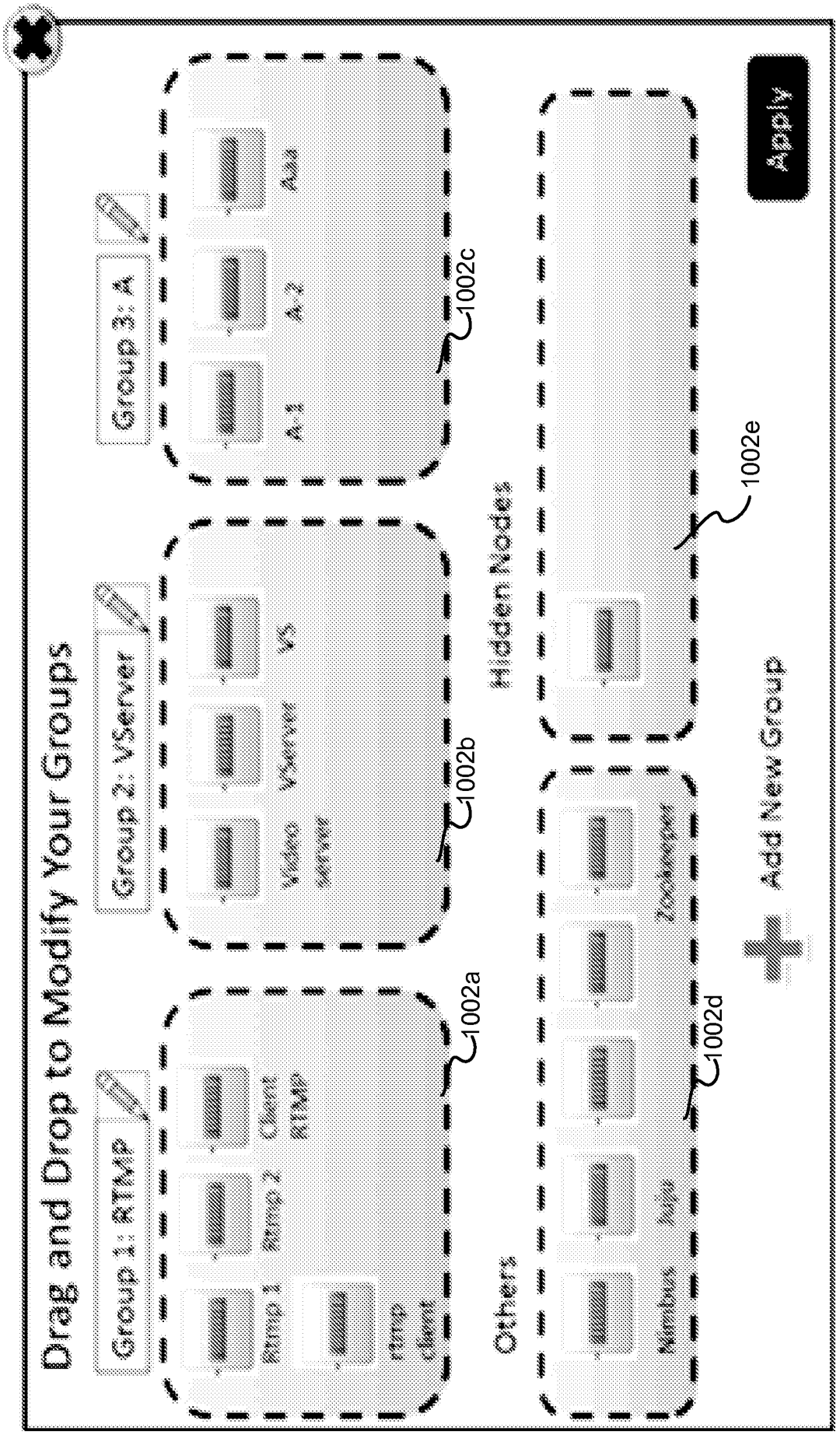

Drag and Drop to Modify Your Groups
Group 1: RTMP
Group 2: VServer
Group 3: A
Hidden Nodes
Others
Add New Group
Apply
1002a
1002b
1002c
1002d
1002e

FIGURE 10

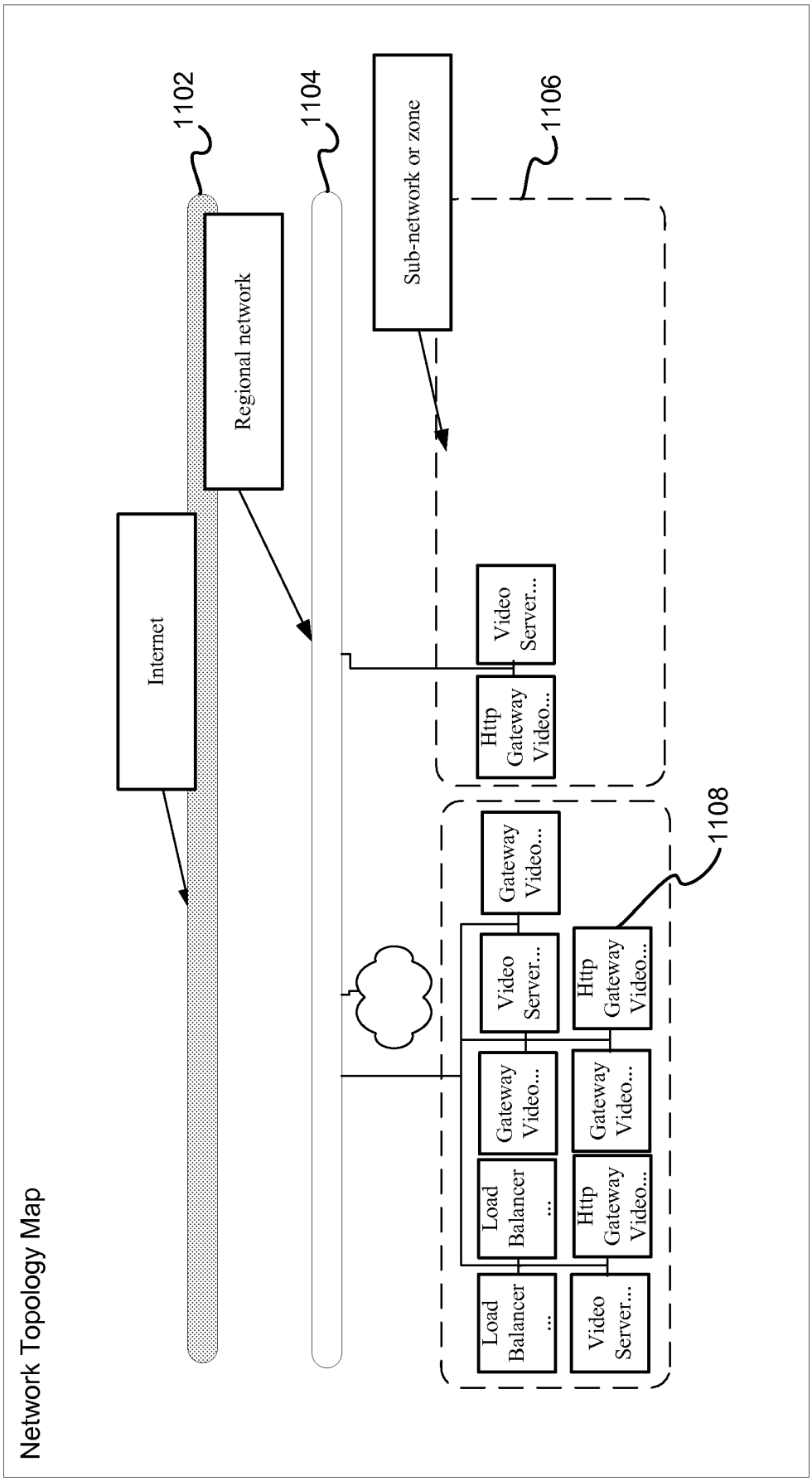

Network Topology Map
Internet
Regional network
Sub-network or zone
1102
1104
1106
1108
Video Server...
Http Gateway Video...
Gateway Video...
Video Server...
Http Gateway Video...
Load Balancer ...
Gateway Video...
Gateway Video...
Load Balancer ...
Http Gateway Video...
Video Server...

FIGURE 11

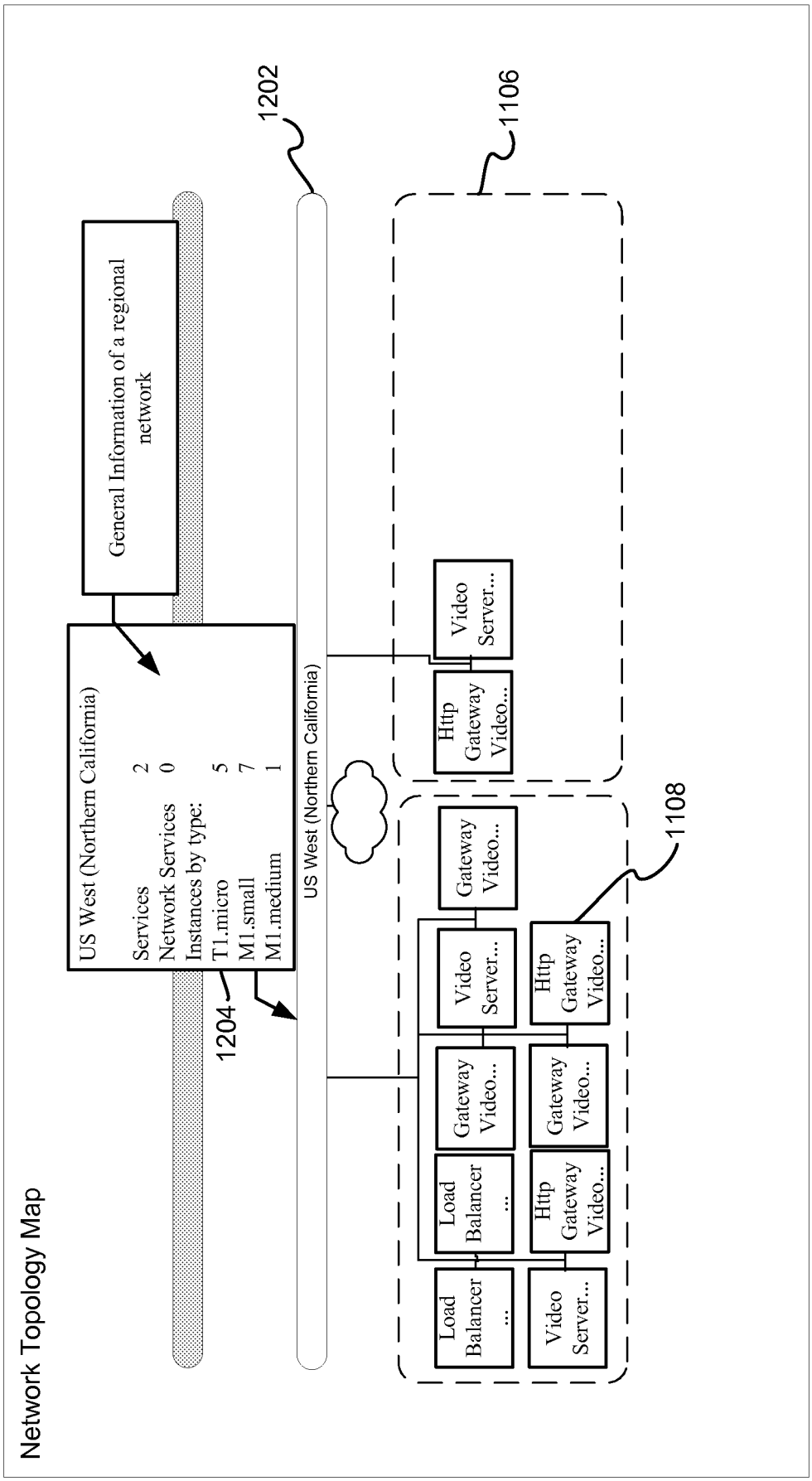

Network Topology Map
General Information of a regional network
US West (Northern California)
Services 2
Network Services 0
Instances by type:
T1.micro 5
M1.small 7
M1.medium 1
1204
1202
US West (Northern California)
1106
1108
Video Server...
Http Gateway Video...
Gateway Video...
Load Balancer ...

FIGURE 12

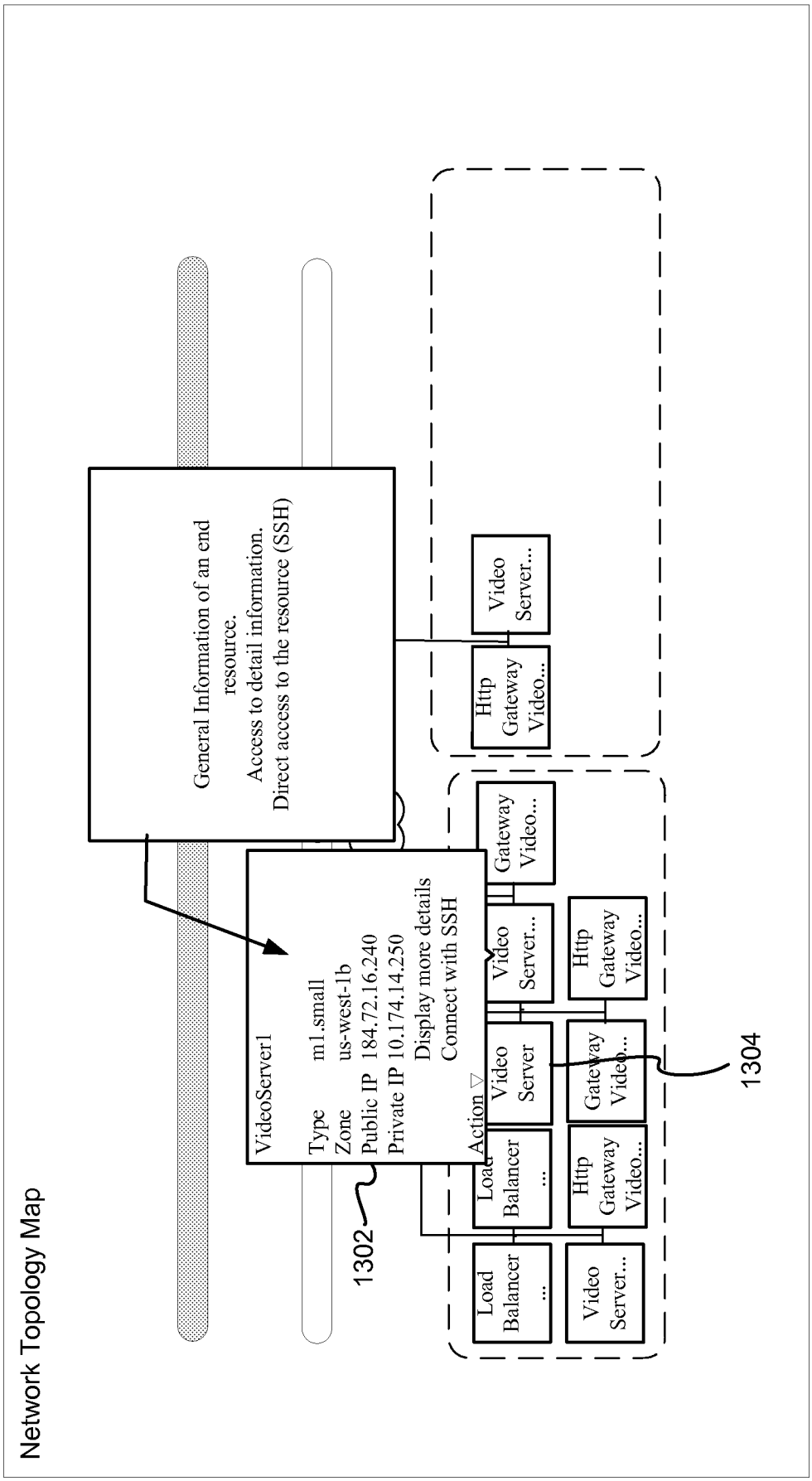

Network Topology Map
General Information of an end resource.
Access to detail information.
Direct access to the resource (SSH)
VideoServer1
Type m1.small
Zone us-west-1b
Public IP 184.72.16.240
Private IP 10.174.14.250
Display more details
Connect with SSH
Action
1302
1304
Load Balancer ...
Video Server...
Http Gateway Video...
Video Server
Gateway Video...
Video Server...
Http Gateway Video...
Load Balancer ...
Http Gateway Video...
Gateway Video...
Video Server...
Http Gateway Video...

FIGURE 13

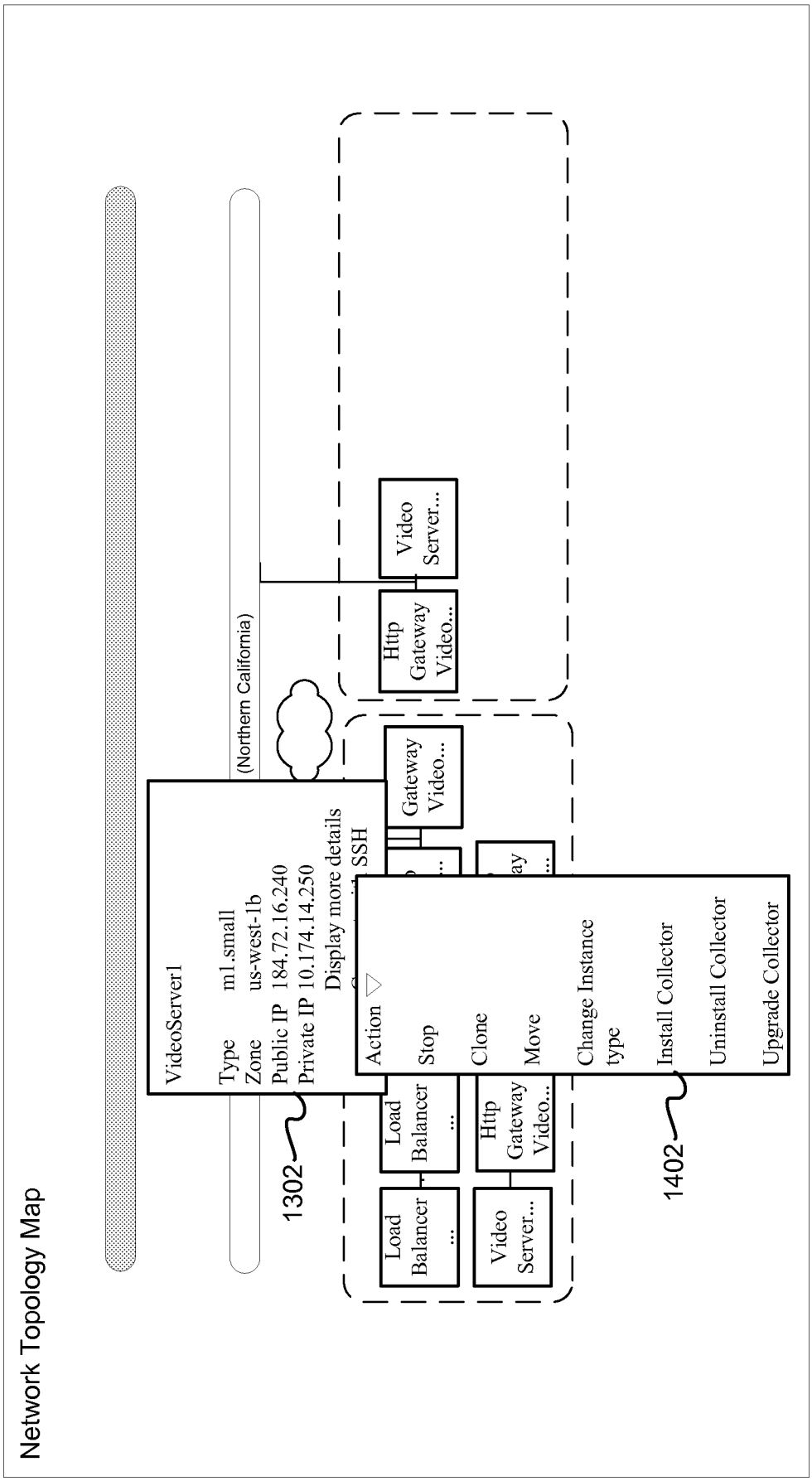

Network Topology Map
(Northern California)
VideoServer1
Type m1.small
Zone us-west-1b
Public IP 184.72.16.240
Private IP 10.174.14.250
Display more details
SSH
Action
Stop
Clone
Move
Change Instance type
Install Collector
Uninstall Collector
Upgrade Collector
1302
1402
Load Balancer ...
Load Balancer ...
Http Gateway Video...
Video Server...
Gateway Video...
Http Gateway Video...
Video Server...

FIGURE 14

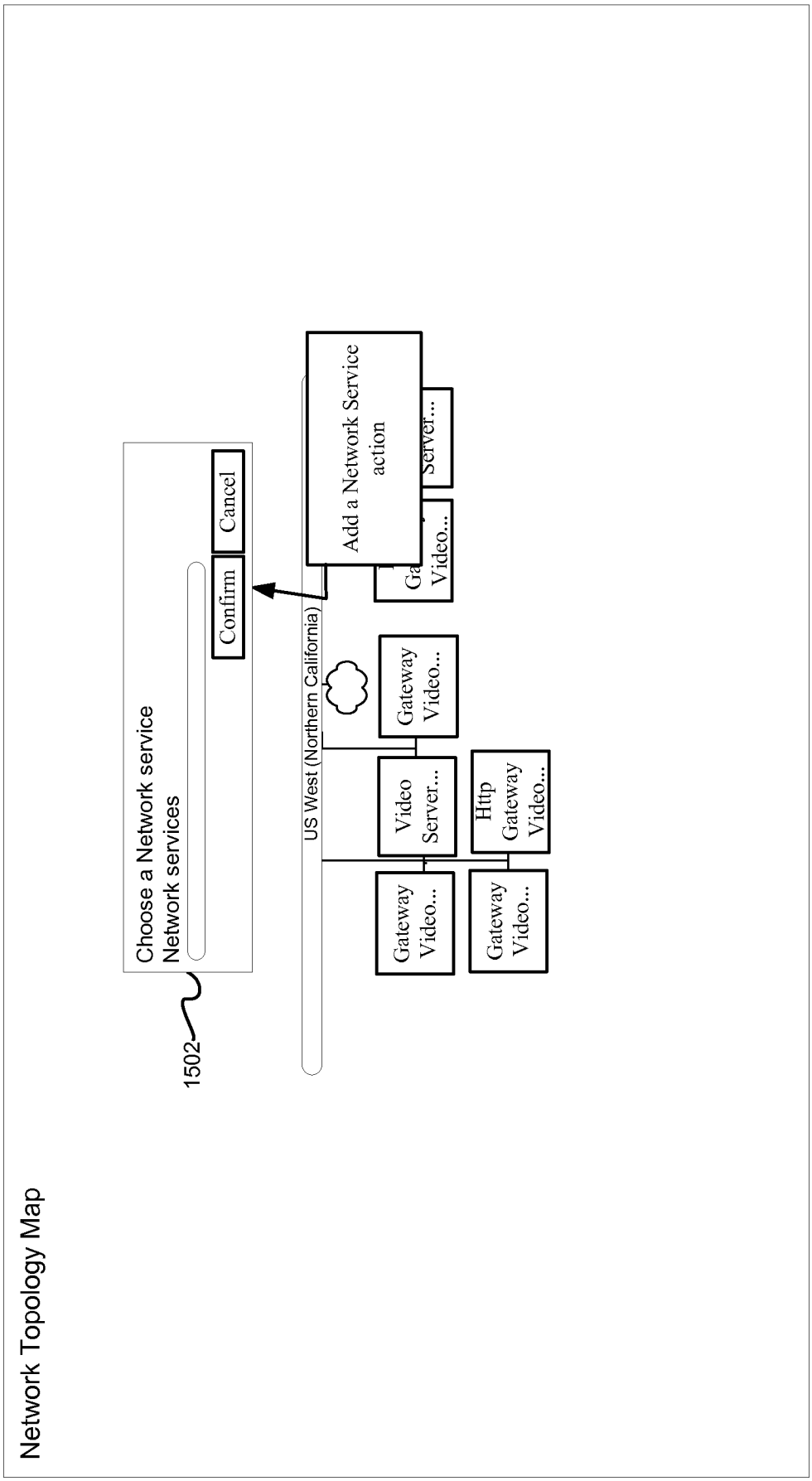

Network Topology Map
Choose a Network service
Network services
Confirm
Cancel
1502
US West (Northern California)
Add a Network Service action
Gateway Video...
Video Server...
Http Gateway Video...
Gateway Video...
Gateway Video...
Gateway Video...
Server...

FIGURE 15

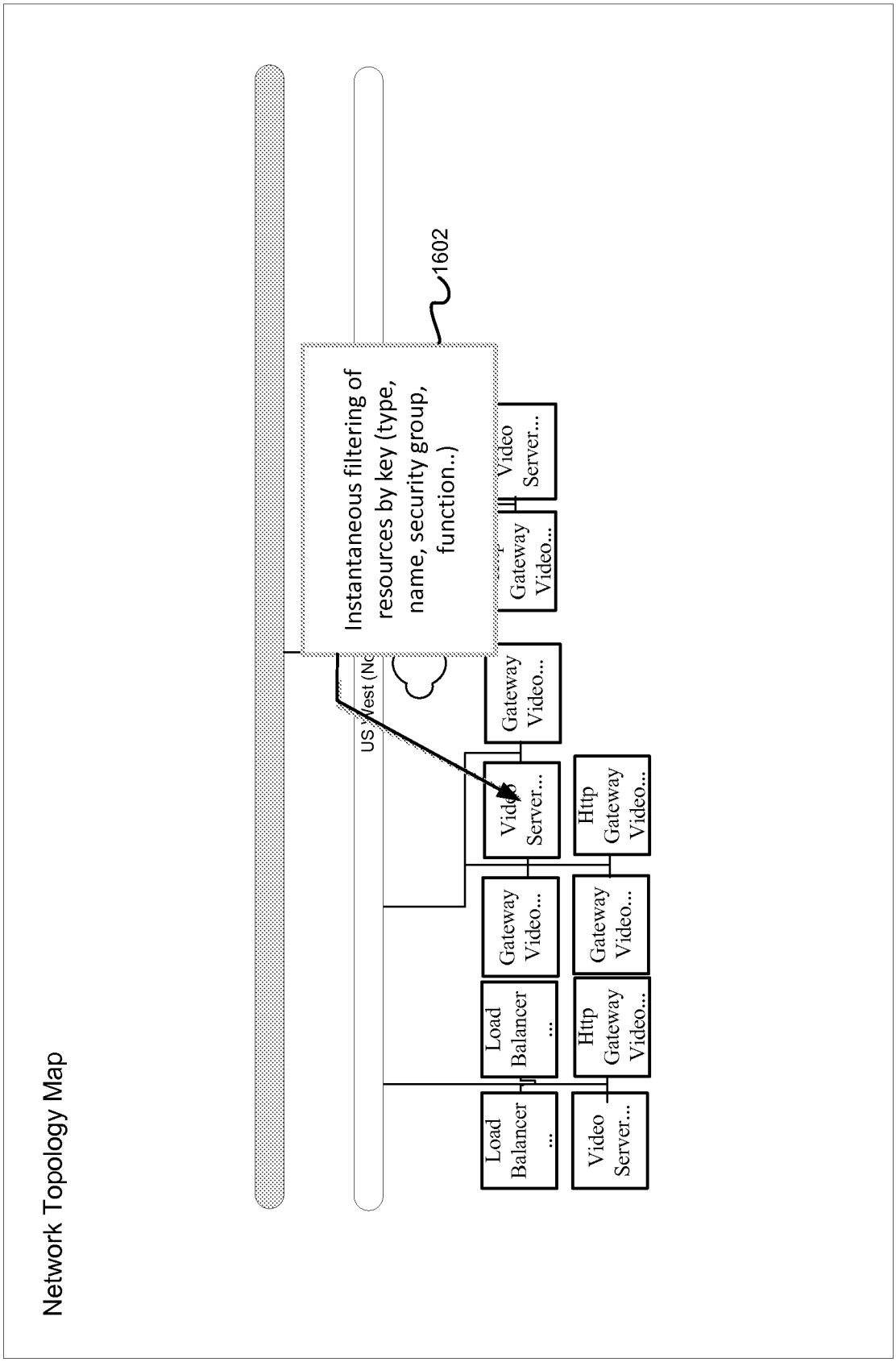

Network Topology Map
Instantaneous filtering of resources by key (type, name, security group, function..)
1602
US West (Nc
Load Balancer ...
Load Balancer ...
Video Server...
Video Server...
Gateway Video...
Gateway Video...
Http Gateway Video...
Http Gateway Video...
Gateway Video...
Gateway Video...
Video Server...
Video Server...

FIGURE 16

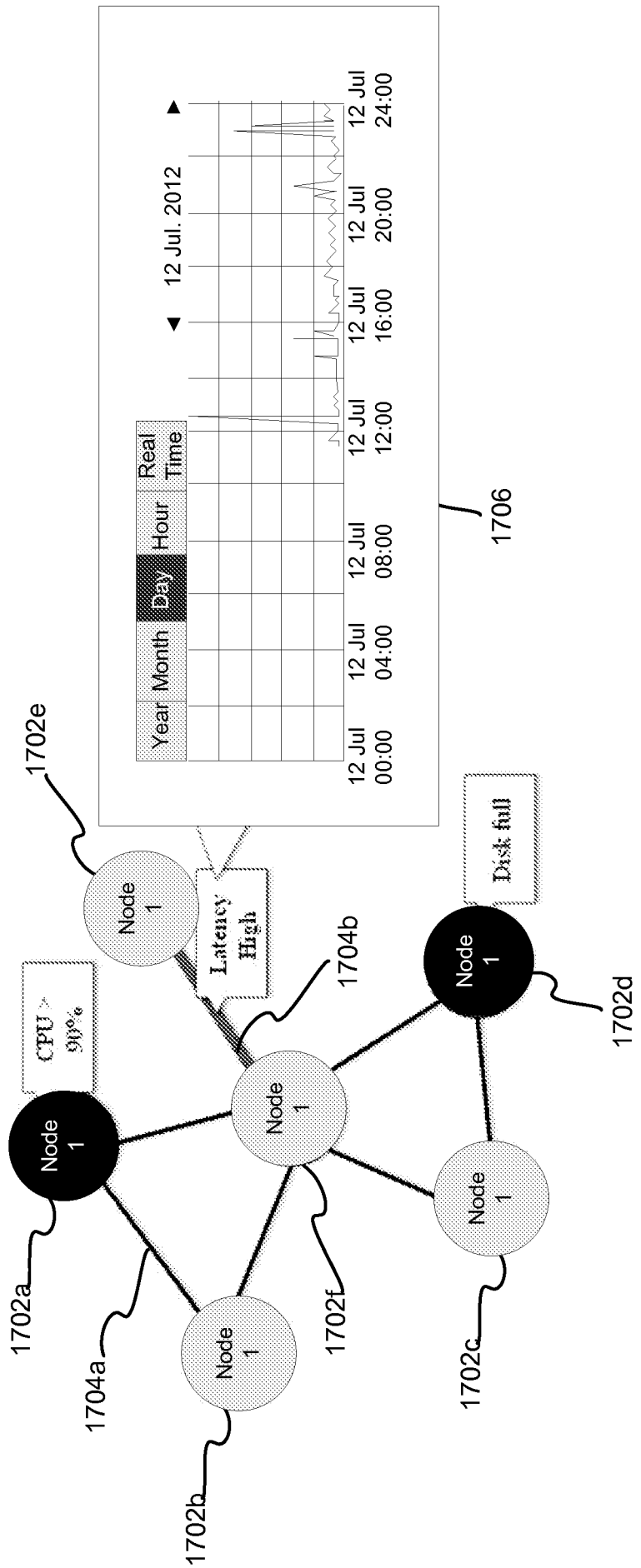

Year
Month
Day
Hour
Real Time
12 Jul. 2012
12 Jul 00:00
12 Jul 04:00
12 Jul 08:00
12 Jul 12:00
12 Jul 16:00
12 Jul 20:00
12 Jul 24:00
1706
1702e
Node 1
Latency High
1704b
CPU > 90%
Node 1
1702a
1704a
Node 1
1702b
Node 1
1702f
Node 1
Disk full
1702d
Node 1
1702c

FIGURE 17

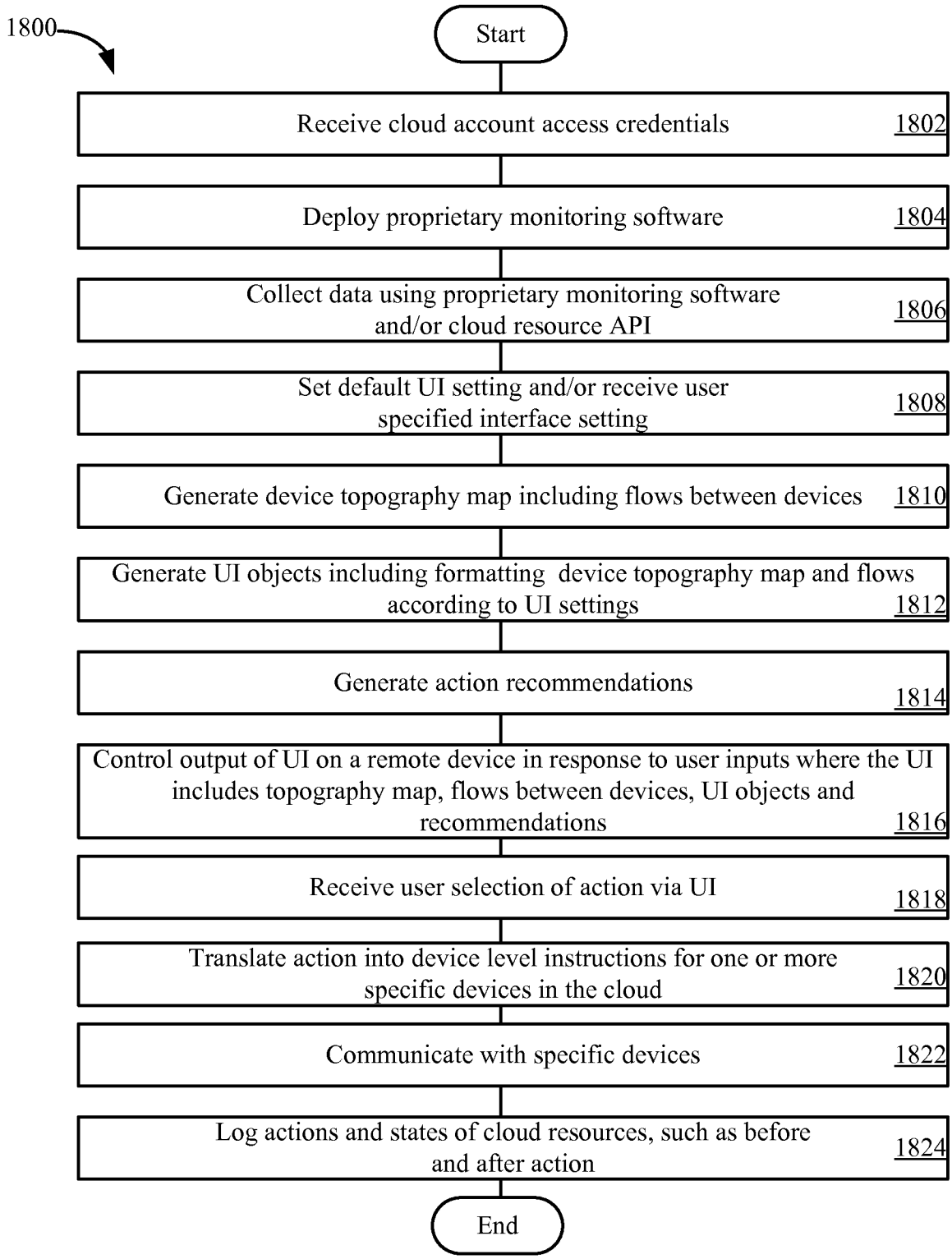

1800
Start
Receive cloud account access credentials 1802
Deploy proprietary monitoring software 1804
Collect data using proprietary monitoring software and/or cloud resource API 1806
Set default UI setting and/or receive user specified interface setting 1808
Generate device topography map including flows between devices 1810
Generate UI objects including formatting device topography map and flows according to UI settings 1812
Generate action recommendations 1814
Control output of UI on a remote device in response to user inputs where the UI includes topography map, flows between devices, UI objects and recommendations 1816
Receive user selection of action via UI 1818
Translate action into device level instructions for one or more specific devices in the cloud 1820
Communicate with specific devices 1822
Log actions and states of cloud resources, such as before and after action 1824
End

FIGURE 18

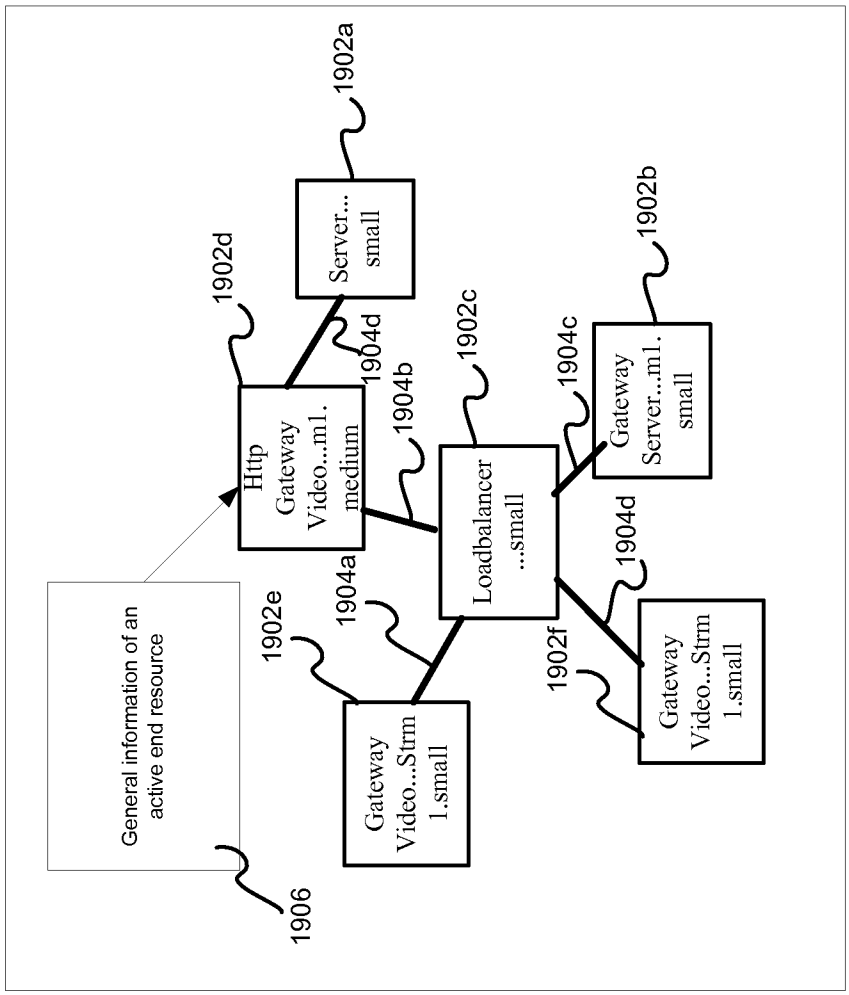

General information of an active end resource
1906
Http Gateway Video...m1. medium
1902d
Server... small
1902a
Loadbalancer ...small
1902c
Gateway Server...m1. small
1902b
Gateway Video...Strm 1.small
1902e
Gateway Video...Strm 1.small
1902f
1904a
1904b
1904c
1904d
1904d

FIGURE 19

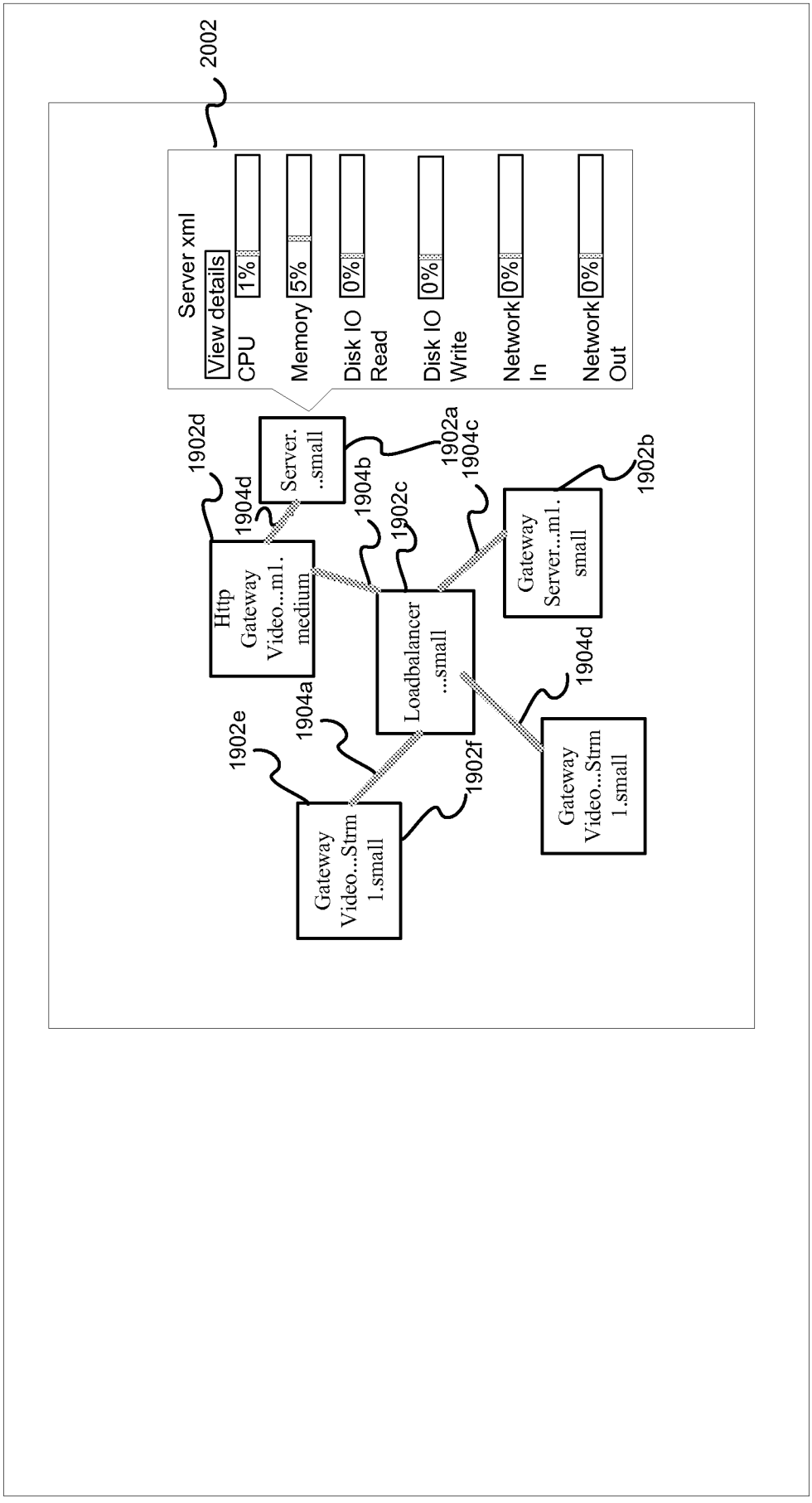

2002
Server xml
View details
CPU 1%
Memory 5%
Disk IO Read 0%
Disk IO Write 0%
Network In 0%
Network Out 0%
1902d
1904d
Server. ..small
1904b
1902a
1904c
1902b
1902c
Http Gateway Video...m1. medium
Gateway Server...m1. small
Loadbalancer ...small
1904a
1904d
1902e
1902f
Gateway Video...Strm 1.small
Gateway Video...Strm 1.small

FIGURE 20

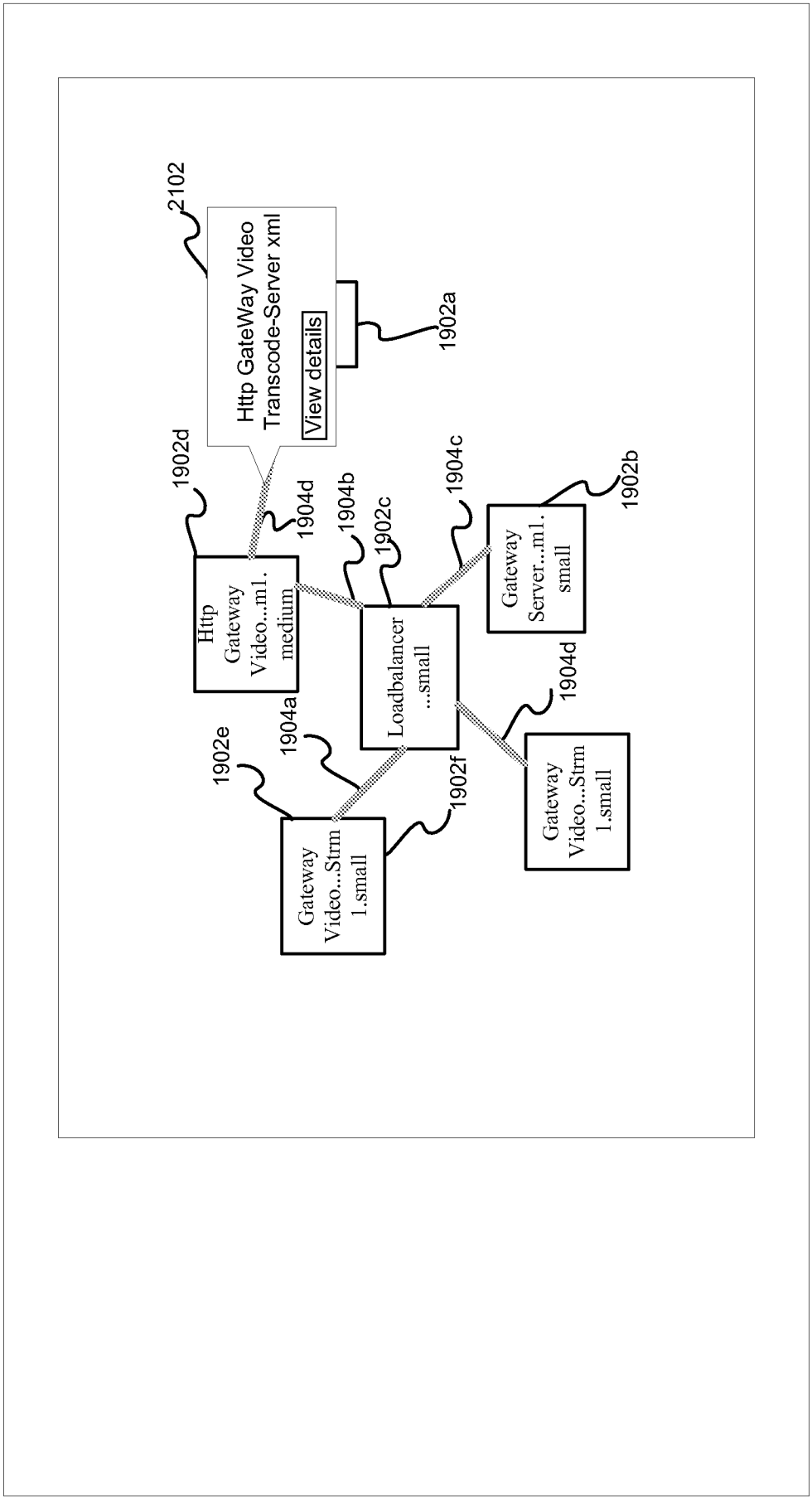

2102
Http GateWay Video Transcode-Server xml
View details
1902a
1902d
1904d
1904b
1902c
1904c
1902b
Http Gateway Video...m1. medium
Loadbalancer ...small
Gateway Server...m1. small
1904a
1902e
1904d
1902f
Gateway Video...Strm 1.small
Gateway Video...Strm 1.small

FIGURE 21

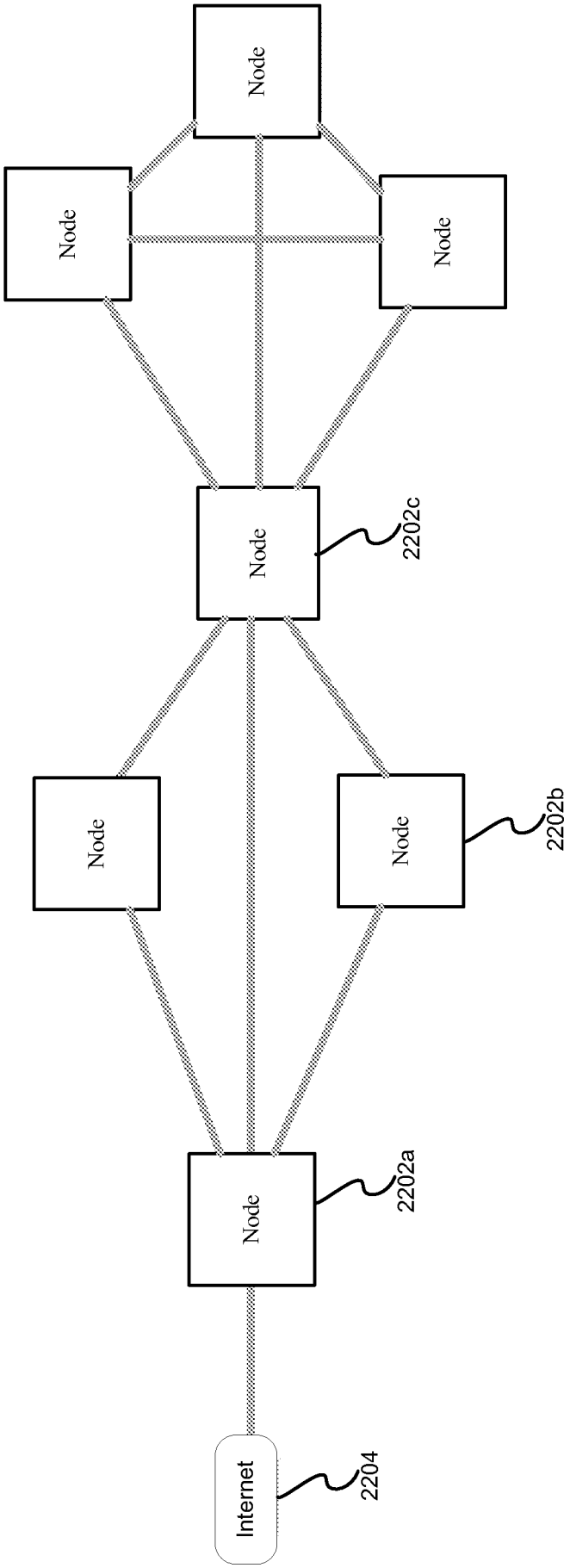

Node
Node
Node
Node
2202c
Node
Node
2202b
Node
2202a
Internet
2204

FIGURE 22

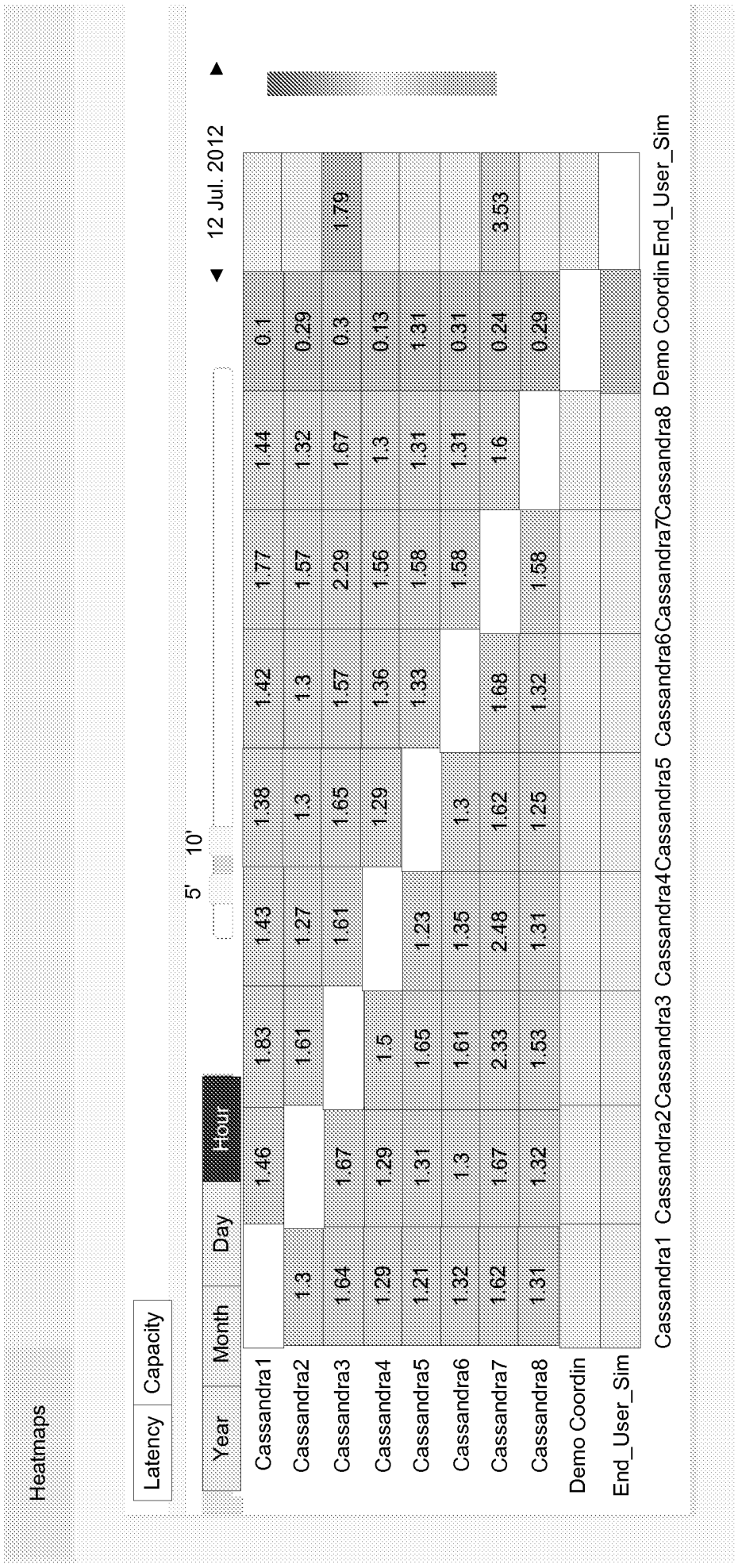

Heatmaps
Latency
Capacity
Year
Month
Day
Hour
5'
10'
12 Jul. 2012
Cassandra1
Cassandra2
Cassandra3
Cassandra4
Cassandra5
Cassandra6
Cassandra7
Cassandra8
Demo Coordin
End_User_Sim
1.46 1.83 1.43 1.38 1.42 1.77 1.44 0.1
1.3 1.61 1.27 1.3 1.3 1.57 1.32 0.29
1.64 1.67 1.61 1.65 1.57 2.29 1.67 0.3 1.79
1.29 1.29 1.5 1.29 1.36 1.56 1.3 0.13
1.21 1.31 1.65 1.23 1.33 1.58 1.31 1.31
1.32 1.3 1.61 1.35 1.3 1.58 1.31 0.31
1.62 1.67 2.33 2.48 1.62 1.68 1.6 0.24 3.53
1.31 1.32 1.53 1.31 1.25 1.32 1.58 0.29
Cassandra1 Cassandra2 Cassandra3 Cassandra4 Cassandra5 Cassandra6 Cassandra7 Cassandra8 Demo Coordin End_User_Sim

FIGURE 23

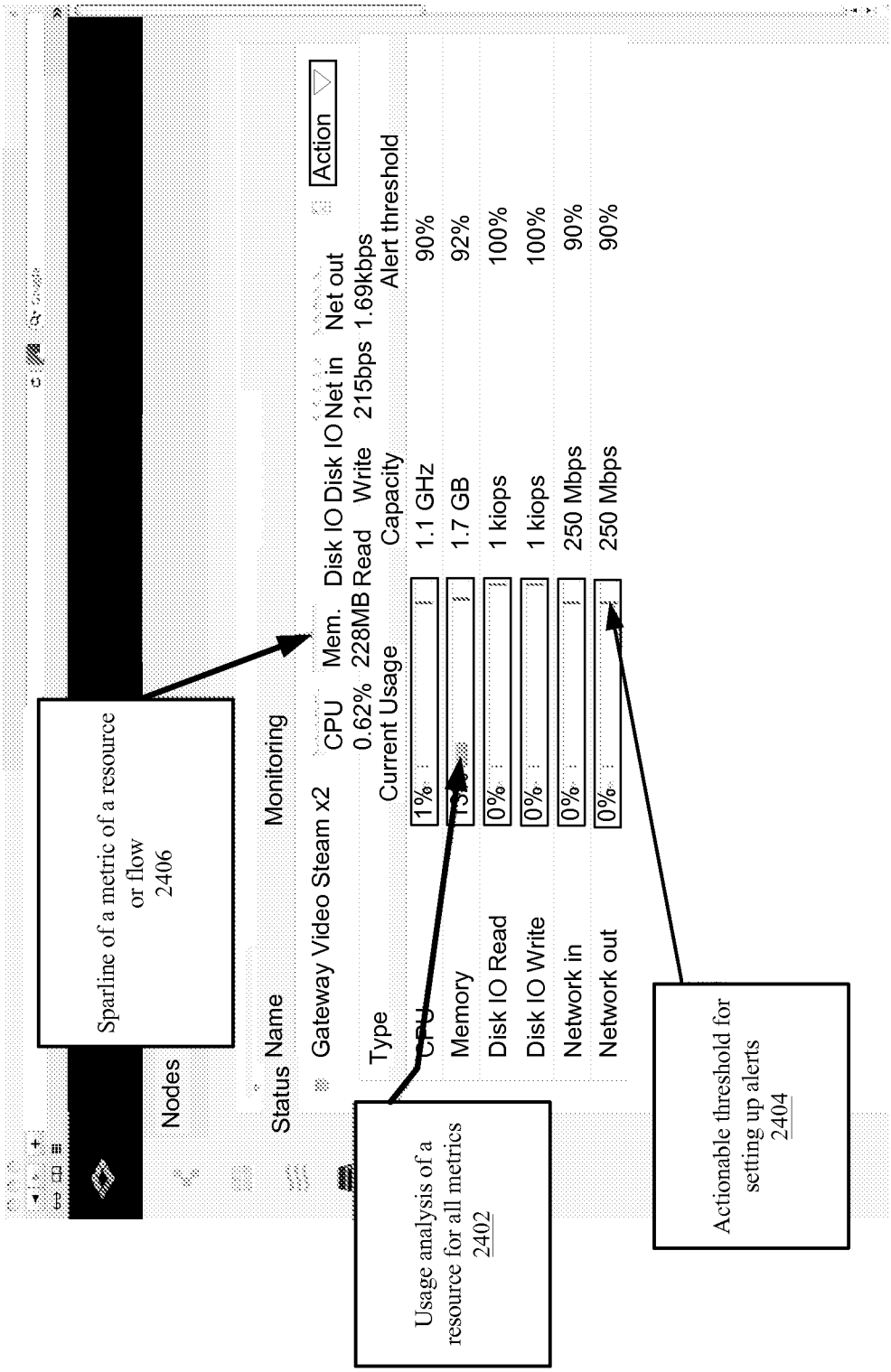

Sparline of a metric of a resource or flow 2406
Usage analysis of a resource for all metrics 2402
Actionable threshold for setting up alerts 2404
Nodes
Status
Name
Monitoring
Gateway Video Steam x2
CPU 0.62%
Mem. 228MB
Disk IO Read
Disk IO Write
Net in 215bps
Net out 1.69kbps
Action
Type
Current Usage
Capacity
Alert threshold
CPU
1%
1.1 GHz
90%
Memory
1.7 GB
92%
Disk IO Read
0%
1 kiops
100%
Disk IO Write
0%
1 kiops
100%
Network in
0%
250 Mbps
90%
Network out
0%
250 Mbps
90%

FIGURE 24

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/011150

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/50(2006.01)i, G06F 17/00(2006.01)i, G06F 15/16(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 9/50; G06F 15/16; G06F 15/173; G06F 21/00; G06F 17/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: cloud, distributed, topology, map, instruction, virtual, resource, processor

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012-0317164 A1 (SHIJUN ZHOU) 13 December 2012 See abstract; claims 1-3, 14, 18, 26-29; and figure 1.	1-25
A	US 2012-0054278 A1 (TALEB TARIK et al.) 01 March 2012 See abstract; claims 1, 4, 5; and figures 1, 2.	1-25
A	US 2012-0110650 A1 (VAN BILJON WILLEM ROBERT et al.) 03 May 2012 See abstract; claims 1, 5, 6, 10; and figure 1.	1-25
A	US 2012-0209980 A1 (JOHNSTON-WATT DUNCAN et al.) 16 August 2012 See abstract; claims 1, 9, 10; and figures 2A, 2B.	1-25

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 March 2014 (26.03.2014)	**31 March 2014 (31.03.2014)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
LEE, Seok Hyung
Telephone No. +82-42-481-5983

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2014/011150

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0317164 A1	13/12/2012	CN 101800762 A	11/08/2010
		EP 2506649 A1	03/10/2012
		WO 2010-148704 A1	29/12/2010
US 2012-0054278 A1	01/03/2012	CN 103329560 A	25/09/2013
		EP 2609750 A2	03/07/2013
		WO 2012-027577 A2	01/03/2012
		WO 2012-027577 A3	31/05/2012
US 2012-0110650 A1	03/05/2012	EP 2583211 A2	24/04/2013
		US 2012-110055 A1	03/05/2012
		US 2012-110056 A1	03/05/2012
		US 2012-110180 A1	03/05/2012
		US 2012-110188 A1	03/05/2012
		US 2012-110636 A1	03/05/2012
		US 2012-110651 A1	03/05/2012
		US 2012-116937 A1	10/05/2012
		US 2012-117229 A1	10/05/2012
		US 2013-060839 A1	07/03/2013
		WO 2011-159842 A2	22/12/2011
		WO 2011-159842 A3	01/03/2012
US 2012-0209980 A1	16/08/2012	EP 2030414 A1	04/03/2009
		GB 0711340 D0	25/07/2007
		GB 2439195 A	19/12/2007
		GB 2439195 B	17/09/2008
		GB 2439195 B8	19/01/2011
		JP 2009-540717 A	19/11/2009
		JP 2009-540717 T	19/11/2009
		JP 2012-043448 A	01/03/2012
		US 2008-016198 A1	17/01/2008
		US 8266321 B2	11/09/2012
		WO 2007-144611 A1	21/12/2007