



(51) International Patent Classification:
Not classified

(21) International Application Number:
PCT/US2020/030121

(22) International Filing Date:
27 April 2020 (27.04.2020)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **FUTUREWEI TECHNOLOGIES, INC.**
[US/US]; 5700 Tennyson Parkway, Suite 600, Plano, Texas 75024 (US).

(72) Inventors: **REN, Da Qi**; 5700 Tennyson Parkway, Suite 600, Plano, Texas 75024 (US). **LIU, Norris**; 5700 Tennyson Parkway, Suite 600, Plano, Texas 75024 (US). **JIANG, Xingyu**; 5700 Tennyson Parkway, Suite 600, Plano, Texas 75024 (US).

(74) Agent: **VIERRA, Larry E.**; Vierra Magen Marcus LLP, 2001 Junipero Serra Blvd., Suite 515, Daly City, California 94014 (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, DJ, DK, DM, DO,

DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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, ST, 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:
— upon request of the applicant, before the expiration of the time limit referred to in Article 21(2)(a)
— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: DYNAMIC RESOURCE TUNING CLOUD SERVICE

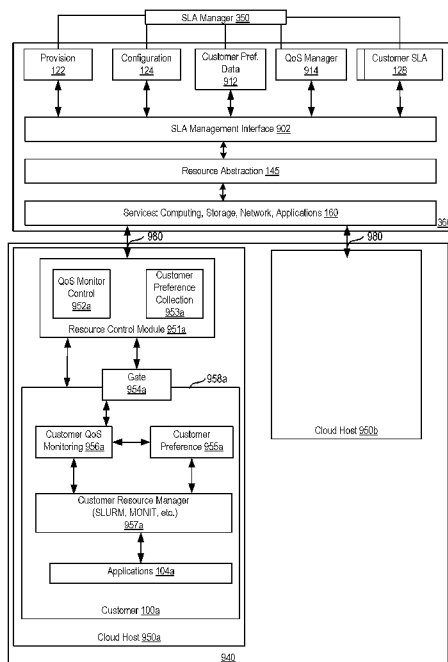


FIG. 9

(57) Abstract: According to one aspect of the present disclosure, an automated method of controlling resources in a cloud computing service is provided. Customer-preference data is defined that identifies, for a customer, a relationship between consumption of one or more first resource types and one or more second resource types. QoS metrics for an application hosted by the computing service for the customer are also defined and provided with the customer-preference to the cloud computing service. On execution of the application, failures in QoS metrics may result in the reallocation of resources which may be performed by the cloud computing service based on the customer-preference data. An automated negotiation between the customer and the computing service may result in a modified customer-preference.



DYNAMIC RESOURCE TUNING CLOUD SERVICE

FIELD

[0001] The disclosure generally relates to the field managing computing resources for a cloud computing service.

BACKGROUND

[0002] A cloud computing provider provides various customers with computing services to give the customer the ability to run applications on the provider's shared computing infrastructure. An advantage of the cloud computing environment is that customers can access additional cloud computing resources when needed and release resources when not needed. When resources are not sufficient to meet customer needs, the cloud provider's services may become unresponsive.

[0003] Service level agreements (SLAs) define a customer's service quality expectations and the obligations of the cloud service provider. The SLA defines multiple Quality of Service (QoS) parameters. Data on measurable capabilities, for example, quality of service, availability and reliability, give the cloud service customer the tools and opportunity to make informed choices and to gain an understanding of the service being delivered. Cloud provider services may be measured by identifying the cloud service properties that have to be measured and their standards of measurement or metrics.

[0004] A metric provides knowledge about characteristics of a cloud property through both its definition (e.g. expression, unit, rules) and the values resulting from the observation of the property. For instance, a customer response time metric can be used to estimate a specific response time property (i.e. response time from customer

to customer) of a cloud service feature. It also provides the necessary information that is needed for to reproduce and verify observations and measurement results.

SUMMARY

[0005] According to one aspect of the present disclosure, there is provided a computer implemented method of controlling resources in a computing service, the computing service having a service level agreement with at least one customer, including: receiving customer-preference data that identifies for the at least one customer a relationship between consumption of one or more first resources and one or more second resources, the customer-preference data constrained by the service level agreement between the customer and the computing service. The computer implemented method of controlling resources also includes receiving a request to execute a set of processes on a plurality of processing resources for the customer. The computer implemented method of controlling resources also includes verifying the service request does not violate that customer's SLA terms. The computer implemented method of controlling resources also includes managing the processing resources allocated to the at least one customer, by determining a resource allocation for the at least one customer; configuring processing resources based on the resource allocation to execute the set of processes through a message passing interface and a processing management interface; monitoring the set of processes through a message passing interface and a processing management interface based on a set of QOS metrics; and based on the monitoring and the customer preference data, and on determining that the QOS metric is not sufficient for the set of processes, automatically modifying the resource allocation and processing resources for the customer. Other embodiments of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the methods.

[0006] Implementations may include one or more of the following methods. The computer implemented method may include any of the foregoing steps and further comprise, for the processing resources allocated to the at least one customer,

launching customer processes, controlling customer processes, and controlling customer processing resources including processing units, processing threads and processing memory. The computer implemented method may include any of the foregoing steps and further comprise providing a resource application programming interface implemented with a processing management interface application programming interface to provide processing management interface services. The computer implemented method may include any of the foregoing steps and further comprise where the customer-preference data identifies, for the customer, the relationship as a rule, the rule based on a metric defining conditions and rules for a measurement of computing service performance. The computer implemented method may include any of the foregoing steps and further comprise where the processing management interface provides each message passing interface process with a number of processing units, and process threads, and a processor group through a group communicator application. method including the foregoing features where the managing resources includes storing a database of customer preference data and resource allocations, and where the managing resources includes adding information to the database and querying information added by applications running on allocated processes to provide the modifying the resource allocation. The computer implemented method may include any of the foregoing steps and further comprise where the modifying includes: initiating an automated negotiation with the computing service using MPI and PMI, requesting a new service level agreement from the computing service, and establishing a new service level agreement using the automated negotiation. The computer implemented method may include any of the foregoing steps and further comprise the QOS metric including one of computing capability or response time. The computer implemented method may include any of the foregoing steps and further comprise the database including storing at least one of processing nodes, computing sockets or processing threads of the allocated resources. Implementations of the described techniques may include hardware, a method or process, or computer software on a computer-accessible medium.

[0007] Another general aspect includes a method of controlling resources for customers in a cloud computing service, each customer having a service level

agreement (SLA) with the cloud computing service, the method including receiving, by the cloud computing service, customer-preference data that identifies, for an application to be executed by the cloud computing service for a customer, a relationship between consumption of a first resource and a second resource. The method of controlling resources also includes receiving a service request to execute a set of processes on processing resources for an application for the customer. The method of controlling resources also includes managing the service level agreement by verifying that the service request does not violate that customer's SLA terms. The method of controlling resources also includes managing resources for the customer, by determining a resource configuration of processing resources for the customer; configuring a set of cloud service processing resources to implement the resource configuration to execute the set of processes using a message passing interface and a processing management interface; monitoring the set of processes based on a set of quality of service (QOS) metrics; and based on the monitoring and the customer preference data, and a determination that one of more of the set of QOS metrics is not sufficient for the set of processes, automatically modifying the set of cloud service processing resources for the customer. Other embodiments of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the methods.

[0008] Implementations may include a method may comprise a method having any of the foregoing steps where managing the SLA includes replying to interleaved polling of QOS metrics responsive to customer requests. The method may comprise a method having any of the foregoing steps and may also include scheduling resource usage to provide QOS for users in a shared environment and resources efficiency. The method may comprise a method having any of the foregoing steps where the customer-preference data identifies, for a customer entity, the relationship as a rule, the rule based on a metric defining conditions and rules for a measurement of computing service performance. The method may comprise a method having any of the foregoing steps where the processing resources include at least computing resources including processing units, processing threads and processing memory. The method may

comprise a method having any of the foregoing steps where the method further includes: receiving a request to begin an automated negotiation with the cloud computing service, negotiating a new service level agreement using the automated negotiation, generating the new service level agreement from the cloud computing service, and establishing the new service level agreement using the automated negotiation. The method may comprise a method having any of the foregoing steps where the QOS metric includes one of computing capability or response time. Implementations of the described techniques may include hardware, a method or process, or computer software on a computer-accessible medium.

[0009] One general aspect includes a processing device operable in a computing service, including: a non-transitory memory storage including instructions; and one or more processors in communication with the memory storage, where the one or more processors execute the instructions. The one or more processors execute the instructions to receive customer-preference data that identifies for a customer a relationship between consumption of one or more first resource types and one or more second resource types, the customer-preference data constrained by a service level agreement between the customer and the computing service. The one or more processors execute the instructions to receive a request for execution of an application from the customer by the computing service; and manage the service level agreement and resources for the customer through a set of MPI and PMI processes. The one or more processors execute the instructions to manage resources by determining that the request does not violate that customer's service level agreement terms, and that the service level agreement service levels are maintained; determining resource configurations for the at least one customer; configuring a set of computing service resources to implement the resource configurations to execute the set of processes through a message passing interface and a processing management interface; monitoring the set of processes based on a set of quality of service (QOS) metrics; and based on the monitoring and the customer preference data, and a determination that one or more of the set of QOS metrics is not sufficient for the set of processes, automatically modifying the set of computing resources for the customer.

[0010] Implementations may include a device having any of the foregoing features where the customer-preference data identifies a relationship for an application executed by the computing service on behalf of the customer. The device may include a device having any of the foregoing features where the customer-preference data identifies, for the customer, the relationship as a rule, the rule based on a metric defining conditions and rules for a measurement of computing service performance. The device may include a device having any of the foregoing features where at least one of the one or more first resource types includes processor allocation. The device may include a device having any of the foregoing features where the resource is thread, core or socket allocation. device may include a device having any of the foregoing features where the one or more processors execute the instructions to: initiate an automated negotiation with the computing service, request a new service level agreement from the computing service, and establish a new service level agreement using the automated negotiation. The device may include a device having any of the foregoing features where the QOS metric includes one of computing capability or response time. The device may include a device having any of the foregoing features where the resource includes one of computing nodes, cores, or threads. Implementations of the described techniques may include hardware, a method or process, or computer software on a computer-accessible medium.

[0011] One general aspect includes a non-transitory computer-readable medium storing computer instructions for one or more processors executed by a cloud computing provider, that when executed by one or more processors, cause the one or more processors to: receive, by the cloud computing provider, customer-preference data that identifies, for an application to be executed by the cloud computing provider for at least one customer, a relationship between consumption of a first resource type and a second resource type; receive a service request to execute a set of processes on cloud computing provider processing resources for the customer; and manage a service level agreement and resources for the customer through a set of processes via message passing interface and a processing management interface, the processes including: The non - transitory computer - readable medium storing computer instructions also includes managing the service level agreement and

resources by: determining that the request does not violate that customer's service level agreement terms, and that the service level agreement service levels are maintained; determining cloud computing service processing resource configurations for the at least one customer; configuring a set of cloud computing resources to implement the resource configurations to execute the set of processes through a message passing interface and a processing management interface, the processors identifying each processing node, processing core, processing thread with a unique identifier; monitoring the set of processes based on a set of QOS metrics; and based on the monitoring and the customer preference data, and a determination that one or more of the set of QOS metrics is not sufficient for the set of processes, automatically modifying the set of computing resources for the customer.

[0012] Implementations may include the non-transitory computer-readable medium having any of the foregoing features where the customer-preference data identifies, for a customer entity, the relationship as a rule, the rule based on a metric defining conditions and rules for a measurement of cloud computing provider performance. The non-transitory computer-readable medium having any of the foregoing features and further where the metric includes one of computing capability or response time. The non-transitory computer-readable medium having any of the foregoing features and further where at least one of the one or more first resource types includes processor allocation. The non-transitory computer-readable medium having any of the foregoing features and further where the resource includes one of sockets, cores, or threads. The non-transitory computer-readable medium having any of the foregoing features and further where the one or more processors to further perform the steps of: receiving a request to begin an automated negotiation with the cloud computing service, negotiating a new service level agreement using the automated negotiation, generating the new service level agreement from the cloud computing service, and establishing the new service level agreement using the automated negotiation.

[0013] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not

intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter. The claimed subject matter is not limited to implementations that solve any or all disadvantages noted in the Background.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] Aspects of the present disclosure are illustrated by way of example and are not limited by the accompanying figures for which like references indicate the same or similar elements.

[0015] FIG. 1 is a block diagram of a computing environment suitable for implementing the present technology.

[0016] FIG 2A is a flowchart illustrating a method performed by a cloud service customer in accordance with the technology.

[0017] FIG. 2B is a flowchart illustrating a method performed by a cloud service provider in accordance with the technology.

[0018] FIG. 3 is a diagram illustrating the relationship between components of a customer-preference, a resource and a metric.

[0019] FIG. 4 is a UML diagram of a cloud service metric and its relationship to a customer-preference.

[0020] FIG. 5A is an illustration of a customer-preference table used in defining a customer-preference.

[0021] FIG. 5B is an illustration of a metric table used in defining a customer-preference.

[0022] FIG. 5C is an illustration of a rule table used in defining a customer-preference.

[0023] FIG. 6 is a flowchart illustrating a method performed by a cloud service provider for a job to be performed for a cloud service customer.

[0024] FIG. 7 is an overview of the negotiation process between a computing service provider and a cloud service customer's computing device relative to each party.

[0025] FIG. 8 is a flowchart illustrating a method performed by a cloud service customer to negotiate a change in service.

[0026] FIG. 9 is a block diagram of components utilized by cloud service customer and a cloud service provider to implement the present technology.

[0027] FIG. 10 is a flowchart illustrating the use of a message passing interface to implement the customer-preference.

[0028] FIG. 11A is a depiction of node management in a cloud service provider using the SLURM cluster management and job scheduling architecture.

[0029] FIG. 11B is a depiction of cluster partitioning using the SLURM cluster management and job scheduling architecture.

[0030] FIG. 11C is a depiction of CPU organization in a quad-core processor using the SLURM cluster management and job scheduling architecture.

[0031] FIG. 12 is a table depicting CPU management using the SLURM cluster management and job scheduling architecture for one cluster.

[0032] FIG. 13 is a table depicting partition management using the SLURM cluster management and job scheduling architecture for one cluster.

[0033] FIG. 14 is a table depicting an exemplary CPU allocation.

[0034] FIG. 15 is a depiction of a computing environment which may be used to implement the present technology.

DETAILED DESCRIPTION

[0035] Technology is provided to allow customers of a cloud services provider to control resource provisioning for the customer using negotiated customer-preference data. QoS in cloud computing is defined in terms of allocating resources to the application that guaranties a service level along dimensions such as performance, availability and reliability. As a cloud services provider performs QoS for the customer, for each request that a customer makes, resources of the provider are allocated or deallocated. When the provided resources result in a measurable insufficiency in performance, as determined by a customer metric, customer-preference data which defines a relationship between resources can be used by the provider to reconfigure resources for the customer. If the reconfigured resources are not sufficient to meet performance objectives, a new SLA may be negotiated via a negotiation process between the customer and the provider.

[0036] In the present technology, customer-preference data is used to define preferences for an application. A resource manager of the cloud service provider may then adjust resources allocated to the customer. The cloud resources managed by the resource manager may include computing, storage and network components that have limited availability. By using different algorithms and approaches, a resource manager performs resource assessment, tracking, and allocation, prevents resource leaks, and terminates access to resources that have been acquired but not released after use, thereby reclaiming resources.

[0037] FIG. 1 is a block diagram of a present cloud service environment including a cloud services provider 190 and cloud services customers resources 100a and 100b that are hosted by 190. The cloud services provider 190 and the hosted computing environments of customer resources 100a, 100b may communicate via a public or private network connection 180. As illustrated in FIG. 1, a cloud services provider 190 may include SLA manager 110, physical resources 140 and a resource manager 120. Physical resources 140 are provided by the cloud services provider, and include various networking, processors and storage equipment. Each of the processors may

be a central processing unit (CPU). The physical resources 140 may comprise a scalable and elastic pool of sharable physical processors or virtual processing resources which may be managed by the resource manager 120. The physical resources 140 may include customer virtual machines which virtualize services on the physical resources 140 to provide customer resources 100a, 100b. Using the physical resources, the cloud services provider 190 provides services 160 which may include, by way of example, dedicated servers, operating systems, networks, software, applications and data storage.

[0038] The physical resources 140 are controlled by resource manager 120. The system components of the resource manager support the cloud services provider 190 in arrangement, coordination and management of physical resources 140 in order to provide cloud services to consumers. The resource manager 120 includes a provisioning controller 122, a configuration controller 124, a QoS manager 130, and customer SLA data 128. The QoS manager 130 and customer-preference data 126 are constrained by customer SLA data 128.

[0039] Shown in FIG. 1 are two sets of customer resources 100a and 100b. While only two sets of customer resources 100a and 100b are illustrated, it will be understood that any number of customers may connect to the cloud services provider 190. Each set of customer resources 100a and 100b includes its own set of customer requirements for customer resource utilization 102a, 102b and customer-data and applications 104a, 104b.

[0040] Cloud service customers can request services and resources be added to their set of resources 100a, 100b, and the cloud services provider 190 controls resources 140 which are utilized by the cloud service customer. Different service models may affect a customer's request over the computing resources 140. In a Software as a Service (SaaS) model, the cloud services provider 190 deploys, configures, maintains and updates the operation of software applications on the computing resources 140 infrastructure so that the services are provisioned at the expected service levels to cloud consumers. The cloud service provider in a SaaS model assumes most of the responsibilities in managing and controlling the

applications and the infrastructure, while the consumers have limited administrative control applications. In a Platform as a Service (PaaS) model, the cloud services provider 190 manages the computing resources 140 infrastructure for the platform and runs the software that provides the components of the platform, such as runtime software execution stack, databases, and other middleware components. In an Infrastructure as a Service, (IaaS) model, the cloud services customers 100a, 100b acquire the physical computing resources underlying the service, including the servers, networks, storage and hosting infrastructure. The cloud services provider 190 runs software necessary to make computing resources available to cloud services customers through a set of service interfaces, such as virtual machines and virtual network interfaces. The cloud service customer can then use these computing resources, such as a virtual computer, for their fundamental computing needs.

[0041] The present technology provides a customer-preference data set that allows the cloud services provider to enable each of the aforementioned models.

[0042] FIG 2A is a flowchart illustrating a method performed by a cloud service customer in accordance with the technology. The method provides customer resource usage data to a cloud service provider that allows the cloud service provider to better adjust resources for the cloud service customer. At 210, the cloud service customer defines customer resource usage data that identifies a relationship between consumption of one or more first resource types and one or more second resource types.

[0043] At 220, the cloud service customer will report resource usage of the application to hosted by the cloud service provider. As described below, the QoS metrics provide a standard of measure for resource performance. At 230, the cloud service customer will monitor the application executing on the cloud service based on data for the metrics. Data for the metrics may be gathered by the cloud services provider. At 240, based on the monitoring, the cloud service customer can request one or more resource types from cloud service provider for the application.

[0044] Figure 2B is a flowchart illustrating a method performed by a cloud service provider. At step 250, the cloud services provider receives customer-preference data that identifies, for the customer application, a relationship between consumption of the first resource type and a second resource type. Again, the resource types may include multiple types of the first resource and multiple types of second resources. At step 260, the cloud service provider determines a consumption level of the first resource type of the application to be executed for the cloud services customer. At step 270, the cloud service provider allocates or deallocates one or more resources of the second resource type based on the identified relationship defined by the cloud service customer. At step 280, the cloud service provider monitors the application based on the customer specified QOS metrics and can provide the monitoring data to the cloud service customer.

[0045] FIGs. 2A and 2B reflect methods that may be performed electronically by processing devices using code, stored in memory associated with the processing device, which causes the processor to execute the methods. Using the customer preference data and the constraints (and goals) defined in a service level agreement, the cloud service processing devices can react quickly to computing needs of customers by altering cloud computing resources allocated to a customer.

[0046] FIG. 3 is a diagram illustrating the relationship between components of customer-preference data (customer-preference rules 325 and customer-preference parameters 335), a resource 310 and a metric 300. A metric 300 may be defined as a standard of measurement characterizing the conditions for performing a measurement of QoS, and for understanding the results of the measurement. As applied, a metric is applied within a given context that require a specific property to be measured, at a given time, for a specific objective. A resource 310, as noted above, may be a computing resource or a service resource provided by the cloud service provider. A scenario 320 represents a particular use case of the cloud service provider for which services may be measured. The scenario 320 can determine the metric to be used for a particular instance of a cloud service. The metric 300 relies on abstract metric definitions that are related to a selected cloud service property. The customer-

preference observation 330 is a measurement based on a metric, at a point in time, on a measurement target. The customer-preference observation 330 of the cloud service through the metric will result in measurement results that can be applied to change the customer-preference parameters. The customer-preference scenario 320 has a direct correlation to the customer-preference rules 325 in use for the application. Customer-preference observations 330 comprise the feedback used by the customer based on customer-preference parameters. For example, there may be a measurable quantification of response time for applications. Customer-preference parameters 335 comprise the measurable factor making up the elements of a metric. As illustrated in FIG. 3, metric 300, customer preference scenario 320 and observation 330 are hosted by SLA manager 350 and resource 310, rules 325 and parameters 335 are hosted by resource manager 360 on the cloud service provider side.

[0047] The customer-preference is managed by the customer by updating associated customer-preference parameter(s) 335 and customer-preference rule(s) 325. Resources 310 are linked to the customer-preference metrics 300, and specific values to the rule(s) and parameter(s) are defined and updated.

[0048] FIG. 4 is a UML diagram of a cloud service metric and its relationship to a customer-preference. The metric is defined by classes which include an abstract metric 460 which includes, for the abstract metric, a rule definition 440, and a parameter definition 450. An instantiated metric includes a metric 420, a metric rule 420 and a metric parameter 430.

[0049] In general, an abstract metric is an abstract standard of measurement used to assess a cloud service property to be observed. The standard of measurement describes what the result of the measurement means, but not how the measurement was performed. The abstract metric is not used by itself but is instantiated using a metric. The abstract metric definition (abstract metric 460, a rule definition 440, and a parameter definition 450) is a collection of elements that defines the expression of a specific metric 300 for a given metric category. A cloud service property is property of a cloud service to be observed. A property may be expressed qualitatively or quantitatively.

[0050] The abstract metric 460 includes basic information necessary to understand the measurement of a property to be observed but does not include the additional information (e.g. Context, target) to actually use the metric 300. This includes: a name of the abstract metric. (such as time/duration); a reference ID, which is a unique identifier for the abstract metric; a unit that will be associated with the abstract metric (such as seconds); a scale including information on how the measurement value can be interpreted and the type of operations which can be performed on it; an expression comprising a function used to assemble the underlying abstract metrics, rule definitions and the parameter definitions (e.g. Expression = sum (response time)/n where “response time” is an underlying abstract metric element and “n” is a parameter definition element); a definition specifying a formal description of the abstract metric; and a note for additional information or comments related to the abstract metric.

[0051] Rule definitions 440 are associated with the abstract metric, and each abstract metric 460 may have zero or more rule definitions associated with it. Rule definitions 440 may be part of the expression of an abstract. An abstract metric 460 may have zero or more parameter definitions 450 associated with it. Parameter definitions 450 may be part of the expression of an abstract metric.

[0052] A rule definition element 440 may be used to further constrain some parts of an abstract metric element and indicate possible method(s) for measurement. The rule definition element 400 has attributes which include its name, referenceID, a definition, and notes, with all attributes having the same meaning as those in the abstract metric class.

[0053] The parameter definition 450 defines a parameter needed in the expression of an abstract metric 460 and a parameter definition may be used by more than one abstract metric. The parameter definition 450 has attributes which include a name, referenceID, definition and note, with these attributes having the same meaning as those in the abstract metric class. It also includes a parameter type attribute defining the manner in which the parameter should be interpreted. (e.g. Integer, string)

[0054] The metric class 410 defines an instantiated abstract metric as a standard of measurement for a specific cloud service property. It is based on the abstract metric element, adding the specific parameters, and rules which are required to use the abstract metric. The metric class includes a name, reference id and note attributes.

[0055] A metric 410 is associated with zero or more metric rules 420. These metric rules 420 are an implementation of an abstract metric through its rule definitions association. A metric 410 is associated with zero or more metric parameters 430. These metric parameters 430 are an implementation of an abstract metric through its parameter definitions association.

[0056] The metric rule 420 represents a concrete rule of the metric based on information from the metric's primary abstract metric element. The metric rule includes a note attribute and a value attribute. The value attribute is the value of the rule defined by the associated rule definition. (e.g. a value could be "scheduled maintenance" for the associated rule definition `observation_exclusion`).

[0057] Each metric rule is dependent on a single rule definition 440. This rule definition 440 is selected from the rule definitions of the primary abstract metric element 460 for the metric 410.

[0058] The metric parameter 430 represents a concrete parameter of the metric based on information from the primary abstract metric element 460 for the metric 410. The value attribute is value of the parameter defined by the associated parameter definition (e.g. a value could be 30 for the associated parameter definition `measurement_timeframe`). The note attribute may contain additional information or comments related to the metric parameter.

[0059] A metric parameter 430 is dependent on a single parameter definition 450. This parameter definition 450 is selected from the parameter definitions of the primary abstract metric element 460 for the metric 410.

[0060] Customer-preference data 490 comprises a set of one or more rules that define, for one or more applications, a preference for adjusting resources based on

the properties defined in the metric. The customer-preference data provides customer-preferences in the allocation and deallocation of resources which are based on measurable qualities of instantiated metrics. Hence, the customer-preference data is dependent upon the instantiated metric to serve a basis for the cloud services provider to allocate or deallocate resources within the context of the SLA, based on the customer defined rules.

[0061] In one aspect, the customer-preference may be provided in a data structure. FIGs. 5A – 5C illustrate one example of customer-preference data in a data structure. It will be recognized that any number of suitable techniques or structures for organizing customer-preference data may be utilized in accordance with the technology.

[0062] FIG. 5A is an illustration of a customer-preference table used in defining a customer-preference. FIG. 5B is an illustration of a metric table used in defining a customer-preference. FIG. 5C is an illustration of a rule table used in defining a customer-preference.

[0063] The customer-preference table illustrated in FIG. 5A may include an entityID, a metric ID and an entity type for each record. As illustrated therein, the customer-preference table may include unique identifiers for various entities within the customer. Preferences may be defined for an application, organization or group within the customer. It should be understood that while the table illustrated one metric for each entity ID, multiple metrics may be defined for each application, organization or group.

[0064] FIG. 5B illustrates a metric table associating the metric IDs from the customer-preference table with rule IDs and resource types. Each rule may be assigned a unique rule ID and associates one or more resources with a rule ID. The resources may be any resource provided by a cloud service provider which is available to a cloud service customer. This description will focus on an implementation using processing capacity, but any resource may be defined in the customer-preference.

[0065] FIG 5C illustrates a rule definition table. For each rule ID, a specific rule is defined. For example, rule 11 increases a second resource if additional resources of a first resource are allocated. Similarly, rule 12 decreases the second resource if the first resource is deallocated. Rule 13 specifies a condition (QoS drops below a threshold) as a precedent to increasing resource 1. It will be understood that the foregoing are examples only.

[0066] FIG. 6 is a flowchart illustrating a method performed by a cloud service provider 190 for an application to be executed for a cloud service customer 100a, 100b, 100c.

[0067] At 610, for any job request from a cloud service customer, at 620, the QOS manager may check the customer-preference data for the cloud service customer to determine the best resource fits for the customer based on the data and as constrained by the SLA. At 630 a QOS manager orders a resource manager to create a resource for the customer of the cloud service provider. At 640, the job (application) requested by the customer executes. At 650, the cloud service provider monitors job performance based on the defined metrics for the customer and may forward the results to the cloud service customer. At 660, a determination is made as to whether not a QOS metric as specified by the customer is not performing adequately. If so, the method returns to step 610 for the next job.

[0068] If the performance of the cloud service provider is not adequate, the customer-preference data is referenced again at 670 to determine whether or not resources may be modified within the scope of the SLA. At 680, resources may be modified by the cloud service provider. This can include increasing CPU frequency or co-locating applications where the resource in question is a deficiency in processing power. The method may continually test whether or not there is a metric violation after adjusting resources at 680

[0069] As discussed above with respect to FIG. 2A, QoS metrics and customer-preference data may be modified if performance by the cloud services provider is not adequate.

[0070] FIG. 7 is an overview of the components used in, and a negotiation process between, a computing service provider and a cloud service customer's computing device relative to each party. FIG. 8 is a flowchart illustrating a method performed by a cloud service customer to negotiate a change in service using the components in FIG. 7.

[0071] As illustrated in FIG. 7, the cloud computing service 190 may include a negotiator process 715, a service and resource management process 725 and a service management process 735. A service level agreement monitor process 710 may also be provided. A cloud computing customer has, as a portion of its organization, a cloud entry point computing device 702, which may be running a cloud negotiator process 704.

[0072] With reference to FIGs. 7 and 8, at 810, a determination is made as to whether the monitored metrics indicate a resource performance which is not acceptable. At 820, a determination is made as to whether not to request modification the SLA to allow for increased allocation of a resource. If so, then at 830, the cloud service customer initiates an SLA negotiation whereby the customer negotiator process 704 initiates a communication with the negotiator process 715 of the cloud service provider 190. At 840, a negotiation response is generated by the negotiator process 715 of the cloud service provider is received by the cloud service customer. At 850, service templates describing the services desired by the cloud services customer are transmitted from the cloud service customer to the cloud service provider service and resource management process 725 negotiator process 704. At 860, service provider templates which define levels of services available from the cloud services provider are sent from the provider 190 to the customer.

[0073] At 870, a negotiation loop comprising a service level request from the cloud services negotiator process 704 and a reply or confirmation received from the service and resource management process negotiate a level of service that is agreeable to both entities until satisfactory agreement is reached. At 880, an agreement offer is transmitted by the cloud service provider to the customer. At 890, the agreement is

executed and at 895, scenario-based metrics and customer service preferences have been established for host application.

[0074] FIG. 9 is a block diagram of the components utilized by cloud service customer and cloud service provider to implement the invented technology. Elements shown in FIG. 9 having like reference numbers to elements in FIG. 1 serve the same purpose and are not discussed further with respect to FIG. 9. In order to implement the present technology, the cloud service provider may include an SLA management interface 902 to the SLA manager 350. The SLA manager ensures that service requests from the customers do not violate that customer's SLA terms, and that the customer's agreed upon service levels are maintained, by using a set of algorithms. Such SLA algorithms include, for example: an interleaved polling algorithm to offer quality of service (QoS) in the upstream direction; and scheduling algorithms to provide quality of service (QoS) for users in a shared environment and at the same time utilize the system resources efficiently. The SLA algorithms ensure the QoS by the cloud service provider, and further include, for example: an Abandonment Rate measure tracking the percentage of calls abandoned while waiting to be answered; an Average Speed to Answer (ASA) monitor: an average time monitor, tracking the average time it takes for a call to be answered by a service desk (usually in seconds); and a Time Service Factor (TSF) measuring the percentage of calls answered within a definite timeframe.

[0075] The SLA manager 350 communicates with the resource manager 360 to monitor, control and ensure service levels as discussed above with respect to FIG. 2B. The resource manager has access to cloud hosts 950a, 950b and physical resources 940 via communication paths 980, as illustrated in FIG 9.

[0076] Resource manager 360 includes a configuration module 124, a provision module 122, a customer-preference-data module 912, QoS manager 914 and customer SLA module 130. The resource manager work with the modules 122, 124, 130, 912 and 914 to access resource rules, configure resources, monitor QoS, manage preference and historical data; and determine resource configurations for customers.

[0077] The SLA management interface 902 gathers all data from cloud customers through Physical resource abstraction module 145 and service module 160.

[0078] Physical resource 940 includes two customer cloud hosts 950a, 950b (Although the details of only cloud host 950a are illustrated, it should be understood that like elements to those discussed with respect to host 950a are contained therein.) The hosts 950a, 950b may comprise a physical server including CPU, memory, storage and OS. The host is controlled by SLA manager 350 and resource manager 360 in the cloud service provider side.

[0079] The host 950a includes a resource-control-module 951a. The host-resource-control module 951a has a QoS monitor and control module 952a, and customer-preference-collection module 953a. Customer-preference-collection module 953a works with the resource manager 360 to detect the resource usage on its host 950a. Note that the resource-control-module 951a is located outside of the set of cloud service customer resources 100a (i.e. the cloud tenant, or the virtual machine).

[0080] A cloud host 950a also includes a set of cloud-service-customer resources (a cloud tenant or virtual machine) 958a. 958a includes a customer-preference module 955a storing the preference data of the local customer, and a customer QoS monitor 956a to monitor the performance inside the resources 958a. The customer resource manager 957a is inside the set of customer resources 958a and includes resource monitoring tools such as SLURM (formerly an acronym for Simple Linux Utility for Resource Management) or MONIT (a free, open-source process supervision tool for Unix and Linux) and similar tools, and is responsible to collect the local data in supporting the customer QoS monitor 956a. Customer resource manager 957a interacts with applications 104a and records what resource pattern is corresponding to what performance, provides such information to customer preference module 955a, and this information is further delivered to resource manager 360.

[0081] Gate 954a is an interface that responsible for the data exchange and communication between the host customer-preference-collector 953a and the

customer QoS Monitor 956a. Gate 954a may be implemented by bridge, network address translation (NAT) or host-only networking.

[0082] A resource management workflow comprises three primary components: (1) SLA Manager 350; (2) control paths 980 (which can be implemented with a message passing interface (MPI) and a process management interface (PMI)) and (3) the resource managers. The resource manager 360 manages a distributed set of processes that controls cloud customers (virtual machines) launching; process control; the environment configuration such as CPU, threads, memories, propagating signals, and the like. Control paths 980 can be implemented with MPI, PMI and a PMI-API. It may be implemented based on the system-specific features, such as Hydra, SLURM, SMPD, to provide PMI services; or, using a typical commodity cluster (e.g. TCP). PMI has wire protocol where data is exchanged through the sockets interface. PMI is a generic tool for any parallel programming model including MPI. PMI can provide each MPI process with resource information about the number of CPUs, and process threads, as well as the resource information of the processor group in an application such as MPI_COMM_WORLD. The resource manager 957a maintains a database of all such information, also allows the SLA manager 350 to interact with the resource manager 957a by adding resource information to the database and querying resource information added by other processes in the applications 104a. Multiple threads can communicate with the host 950a and turn the resource and performance according to a customer's SLA specification.

[0083] A computing cloud can support high-performance computing (HPC) with many advantages. For example, cloud computing provides a resilient and scalable cloud infrastructure to run HPC applications with almost unlimited capacity. HPC system owners deploy their tasks that stress the limits of local HPC infrastructure. Cloud can offer an integrated services suite that includes everything needed to quickly and easily build and manage HPC clusters in the cloud, enabling vertical industries to use them to run the most compute-intensive workloads. These workloads include traditional HPC applications as well as emerging applications.

[0084] HPC over cloud can eliminate the long wait times and lost productivity typically associated with local HPC clusters. Flexible configurations and almost unlimited scalability allow to increase and decrease infrastructure as user workload demands.

[0085] The mainstream of parallel computing is parallelism between computer nodes, and MPI's a major technology for distributed computing on them. MPI was developed for establishing a practical, portable, effective, and flexible standard for messaging. MPI proposes a function interface description based on message passing. MPI can be used to implement the MapReduce function in Hadoop, and the MPI model is widely used in parallel machines, especially distributed storage parallel machines.

[0086] For example, a master process can assign a job to a cloud computing system by sending a message describing the job to a slave process. Another example is a concurrent program that can sort the data in a current process, and then send the sorted data to a neighbor process for a merge operation. Almost all parallel programs can be described using a message passing model.

[0087] PMI allows different process managers to interact with the MPI library in a standardized way. It is widely used in MPICH2 and other derivative MPI implementations, such as MVAPICH2, Intel MPI. It is suitable for modern HPC systems, including a hybrid programming model that combines MPI and threads with scalability and efficient interaction that can implement a large number of cores on a single node.

[0088] The implementation of PMI depends on the system itself. For example, for a typical product process cluster, the PMI library can communicate via a communication path (such as TCP) and the data can pass through a socket interface. The advantage of TCP protocol is that any application is compatible.

[0089] SLA manager 350 stores SLA metrics and works with SLA Management interface 902, together with resource manager 360, to communicate with the resource control module 951a on each host 950a via communication paths 980. This

communication typically requires a contact address, which may be an IP address, a remotely accessible memory segment, or any other interconnect-specific identifier. A resource manager 957a keeps the users and applications managed by itself. It also queries the status and information from remote nodes via control paths through management interface. Each resource manager maintains a separate database and allowed to query for information across databases. It allows SLA managers to exchange their database information and load them into their individual databases.

[0090] Because the SLA manager and resource manager have access to the MPI and PMI interfaces, they can be utilized the QoS manager and customer preference data to dynamically change sets of cloud service computing resources to customers. This improves the speed at which the cloud service provider can respond to changes in customer computing needs, allows the cloud service provider to more accurately manage (release and allocate) its computing resources between customers, and improve the execution of customer processes. This makes the cloud computing service response time more accurate, and its computing resources (processors, dynamic memory, storage, and networking, for example,) more efficient.

[0091] FIG. 10 is a flowchart illustrating the use of a message passing interface (MPI) to implement the customer-preference by allocating resources in a cloud service provider. At 1010, a resource manager in a cloud service provider determines a resource allocation and at 1020 may access an MPI library. At 1030, tasks are launched using MPI and a resource launching mechanism, such as SLURM. SLURM is an open source and scalable cluster management and job scheduling system for large and small clusters. SLURM can allocate exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work and provides a framework for starting, executing, and monitoring work on a set of allocated nodes. At 1040, the cloud service provider waits for the next job request.

[0092] In one embodiment, SLURM is used the resource manager and the technology will be described using this example. It should be understood that the technology is not limited to the use of SLURM as a resource manager and other resource managers may be used.

[0093] FIG. 11A is a depiction an implementation of node management in a cloud service provider using the SLURM cluster management and job scheduling architecture, illustrating how entities are managed. A cloud service provider includes a processing environment 1115 which may comprise physical and/or virtual resources organized into a plurality of compute nodes 1120. SLURM consists of a daemon 1125 running on each compute node and a control daemon 1110 running on a management node. The daemons provide fault-tolerant hierarchical communications. Daemon 1125 provides machine status, job status, remote execution and stream copy functions, and executes on every compute node 1120.

[0094] A controller 1110 interfaces with each node 1120 in the processing environment 1115 via daemon 1125. The controller 1110 includes a node manager, partition manager and job scheduler. The controller 1110 orchestrates processing environment activities, including queuing of jobs, monitoring node states, and allocating resources to jobs.

[0095] FIG. 11B is a depiction of cluster partitioning using the SLURM cluster management and job scheduling architecture. As illustrated in FIG. 11B, each cluster 1130 may include compute resource node partitions 1132, 1134, which group compute nodes into logical sets. Partitions may be further grouped into jobs (Job A, Job B, Job C), and job steps (Job Step 1, job Step 2, Job Step 3), which are sets of tasks within a job. The partitions can be considered job queues, each of which has an assortment of constraints such as job size limit, job time limit, users permitted to use it, and the like. Priority-ordered jobs are allocated nodes (Node A through Node I in FIG. 11B) within a partition until the resources (nodes, processors, memory, and the like) within that partition are exhausted.

[0096] FIG. 11C is a depiction of CPU organization in a quad-core processor using the SLURM cluster management and job scheduling architecture. SLURM allows selecting configuration options to manage the use of CPU resources by jobs, steps and tasks. There are four basic steps to manage CPU resources for a job/step: selection of nodes; allocation of CPUs from the selected nodes; Distribution of Tasks

to the selected Nodes; and, optionally, Distribution and Binding of Tasks to CPUs within a Node.

[0097] In one embodiment, customer-preferences are defined in terms of resources comprising compute or processing resources. In SLURM, whole nodes, cores or threads may be allocated as resources. Exclusive (unshared) allocation of CPUs as consumable resources limits the number of jobs/steps/tasks that can use a node concurrently, but it does not limit the set of CPUs on the node that each task distributed to the node can use. Unless some form of CPU/task binding is used (e.g., a task or spank plugin), all tasks distributed to a node can use all of the CPUs on the node, including CPUs not allocated to their job/step. In one embodiment, consumable resources can be configured with task affinity to unshared CPUs, shared CPUs, and/or threads. Task affinity can also be useful when select/linear (whole node allocation) is configured, to improve performance by restricting each task to a particular socket or other subset of CPU resources on a node.

[0098] One example of node and partition configuration is shown in FIGs 12 – 14. FIG. 12 is a table depicting a node cluster used in one embodiment of management and job scheduling. FIG. 13 is a table depicting partition management using the SLURM cluster management and job scheduling architecture for the cluster of FIGS. 12.

[0099] FIG 14 provides an exemplary configuration of a job using 6 CPUs. The objective in this example is to run a job in a single node in the default partition, with core binding to each task and cyclic distribution of tasks to nodes, and block distribution of tasks for CPU binding. Using the default allocation method within nodes (cyclic), one can allocate 3 CPUs on each socket of 1 node. Using the default distribution method within nodes (cyclic), one can distribute and binds each task to an allocated core in a round-robin fashion across the sockets. The table of FIG. 14 shows a possible pattern of allocation, distribution and binding for this job. For example, task id 2 is bound to CPU id 1, as shown in FIG. 14.

[00100] By way of example only, a resource manager using SLURM can allocate whole nodes or cores as consumable resources. This allocation can be: nodes, an allocation with balanced allocation across nodes, an allocation with minimization of resource fragmentation; an allocation with cyclic distribution of tasks to nodes; an allocation with plane distribution of tasks to nodes; an allocation with overcommitment of CPUs to tasks; an allocation with resource sharing between jobs; an allocation with to a multithreaded node, allocating only one thread per core; an allocation with task affinity and core binding; an allocation with task affinity and socket binding; and/or an allocation with task affinity and customized allocation and distribution.

[00101] Hence, a cloud service provider may utilize the present technology to manage CPU allocation and task affinity/binding as the resources available based on the customer-preference.

[00102] For processor (CPU) allocation, customer-preferences (as illustrated in FIG. 5C may include rules such as: If additional resources of nodes are allocated, also allocate additional random access memory; if additional computing resources are allocated, allocate additional unbound nodes; and/or if additional computing resources are allocated, allocate only one thread per core. These customer-preference rules are by way of example only.

[00103] FIG. 15 is a block diagram of a network device 1500 that can be used to implement various embodiments. Specific network devices may utilize all of the components shown, or only a subset of the components, and levels of integration may vary from device to device. Furthermore, the network device 1500 may contain multiple instances of a component, such as multiple processing units, processors, memories, transmitters, receivers, etc. The network device 1500 may include a central processing unit (CPU) 1510, a memory 1520, a mass storage device 1530, and an I/O interface 1560 connected to a bus 1570. The bus 1570 may be one or more of any type of several bus architectures including a memory bus or memory controller, a peripheral bus or the like.

[00104] The CPU 1510 may comprise any type of electronic data processor. The memory 1520 may comprise any type of system memory such as static random-access memory (SRAM), dynamic random-access memory (DRAM), synchronous DRAM (SDRAM), read-only memory (ROM), a combination thereof, or the like. In an embodiment, the memory 1520 may include ROM for use at boot-up, and DRAM for program and data storage for use while executing programs.

[00105] In embodiments, the memory 1520 is non-transitory. In one embodiment where the network device is a component of the cloud service provider, the memory 1520 includes the SLA manager 350, the resource manager, monitoring engine, configuration engine and infrastructure and resource module.

[00106] In the FIG. 15 illustrated embodiment, the network device is a component of the cloud service customer and includes a customer-preference generator 1520A, a QoS monitor 1520B and a job request/allocation engine 1520c. A negotiation engine 1520d may also be included to perform the negotiation process discussed above.

[00107] The mass storage device 1530 may comprise any type of storage device configured to store data, programs, and other information and to make the data, programs, and other information accessible via the bus 570. The mass storage device 1530 may comprise, for example, one or more of a solid-state drive, hard disk drive, a magnetic disk drive, an optical disk drive, or the like. The mass storage device 1530 may include customer-preference data 104 for the customer.

[00108] The mass storage device may also store the any of the components described as being in or illustrated in memory 1520 to be read by the CPU and executed in memory 1520. The mass storage device may comprise computer-readable non-transitory media which includes all types of computer readable media, including magnetic storage media, optical storage media, and solid-state storage media and specifically excludes signals. It should be understood that the software can be installed in and sold with the network device. Alternatively the software can be obtained and loaded into network _ device, including obtaining the software via a disc medium or from any manner of network or distribution system, including, for example,

from a server owned by the software creator or from a server not owned but used by the software creator. The software can be stored on a server for distribution over the Internet, for example.

[00109] The network device 1500 also includes one or more network interfaces 1550, which may comprise wired links, such as an Ethernet cable or the like, and/or wireless links to access nodes or one or more networks 1580. The network interface 1550 allows the network device 1500 to communicate with remote units via the networks 1580. For example, the network interface 1550 may provide wireless communication via one or more transmitters/transmit antennas and one or more receivers/receive antennas. In an embodiment, the network device 1500 is coupled to a local-area network or a wide-area network 1580 for data processing and communications with remote devices, such as other processing units, the Internet, remote storage facilities, or the like.

[00110] For purposes of this document, it should be noted that the dimensions of the various features depicted in the figures may not necessarily be drawn to scale.

[00111] For purposes of this document, reference in the specification to “an embodiment,” “one embodiment,” “some embodiments,” or “another embodiment” may be used to describe different embodiments or the same embodiment.

[00112] For purposes of this document, a connection may be a direct connection or an indirect connection (e.g., via one or more other parts). In some cases, when an element is referred to as being connected or coupled to another element, the element may be directly connected to the other element or indirectly connected to the other element via intervening elements. When an element is referred to as being directly connected to another element, then there are no intervening elements between the element and the other element. Two devices are “in communication” if they are directly or indirectly connected so that they can communicate electronic signals between them.

[00113] Although the present disclosure has been described with reference to specific features and embodiments thereof, it is evident that various modifications and

combinations can be made thereto without departing from scope of the disclosure. The specification and drawings are, accordingly, to be regarded simply as an illustration of the disclosure as defined by the appended claims, and are contemplated to cover any and all modifications, variations, combinations or equivalents that fall within the scope of the present disclosure.

[00114] The foregoing detailed description has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the subject matter claimed herein to the precise form(s) disclosed. Many modifications and variations are possible in light of the above teachings. The described embodiments were chosen in order to best explain the principles of the disclosed technology and its practical application to thereby enable others skilled in the art to best utilize the technology in various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope be defined by the claims appended hereto.

CLAIMS

What is claimed is:

1. A computer implemented method of controlling resources in a computing service, , comprising:

receiving customer-preference data that identifies for the at least one customer a relationship between consumption of one or more first resources and one or more second resources, the customer-preference data constrained by a service level agreement (SLA) between the customer and the computing service as reflected in the customer-preference data;

receiving a request to execute a set of processes on a plurality of processing resources for the customer;

verifying the service request does not violate that customer's SLA terms, and managing the processing resources allocated to the at least one customer, by

determining a resource allocation for the at least one customer;

configuring processing resources based on the resource allocation to execute the set of processes though a message passing interface and a processing management interface;

monitoring the set of processes though the message passing interface and the processing management interface based on a set of QoS metrics;
and

based on the monitoring, determining that the QoS metric is not sufficient for the set of processes;

automatically modifying the resource allocation and processing resources for the customer based on the customer preference data and the determination that the QoS metric is not sufficient.

2. The computer implemented method of claim 1 wherein the managing resources comprises, for the processing resources allocated to the at least one customer, launching customer processes, controlling customer processes, and

controlling customer processing resources including processing units, processing threads and processing memory.

3. The computer implemented method of any of claims 1 through 2 wherein the managing resources comprises providing a resource application programming interface implemented with a processing management interface application programming interface to provide processing management interface services.

4. The computer implemented method of any of claims 1 through 3 wherein the customer-preference data identifies, for the customer, the relationship as a rule, the rule based on a metric defining conditions and rules for a measurement of computing service performance.

5. The computer implemented method of any of claims 1 through 4 wherein the processing management interface provides each message passing interface process with a number of processing units, and process threads, and a processor group through a group communicator application.

6. The computer implemented method of any of claims 1 through 5 wherein the managing resources comprises storing a database of customer preference data and resource allocations, and wherein the managing resources comprises adding information to the database and querying information added by applications running on allocated processes to provide the modifying the resource allocation.

7. The computer implemented method of any of claims 1 through 6 wherein the modifying comprises:

initiating an automated negotiation with the computing service using MPI and PMI;

requesting a new service level agreement from the computing service; and
establishing a new service level agreement using the automated negotiation.

8. The computer implemented method of any of claims 1 through 7 wherein the QoS metric comprises one of computing capability or response time.
9. The computer implemented method of any of claims 1 through 8 wherein the database includes storing at least one of processing nodes, computing sockets or processing threads of the allocated resources.
10. A method of controlling resources for customers in a cloud computing service, the method comprising:
 - receiving, by the cloud computing service, customer-preference data that identifies, for an application to be executed by the cloud computing service for a customer, a relationship between consumption of a first resource and a second resource;
 - receiving a service request to execute a set of processes on processing resources for an application for the customer;
 - managing a service level agreement by verifying that the service request does not violate that customer's SLA terms, and
 - allocating resources for the customer, by
 - determining a resource configuration of cloud service processing resources for the customer;
 - configuring a set of cloud service processing resources to implement the resource configuration to execute the set of processes using a message passing interface and a processing management interface;
 - monitoring the set of processes based on a set of quality of service (QoS) metrics; and
 - based on the monitoring and the customer preference data, and a determination that one of more of the set of QoS metrics is not sufficient for the set of processes, automatically modifying the set of cloud service processing resources for the customer.

11. The method of claim 10 wherein managing the SLA includes:
replying to interleaved polling of QoS metrics responsive to customer requests;
and
scheduling resource usage to provide QoS for users in a shared environment and resources efficiency.
12. The method of any of claims 10-11 wherein the customer-preference data identifies, for a customer entity, the relationship as a rule, the rule based on a metric defining conditions and rules for a measurement of computing service performance.
13. The method of any of claims 10 through 12 wherein the processing resources comprise at least computing resources including processing units, processing threads and processing memory.
14. The method of any of claims 10 through 13 wherein the method further includes:
receiving a request to begin an automated negotiation with the cloud computing service;
negotiating a new service level agreement using the automated negotiation;
generating the new service level agreement from the cloud computing service;
and
establishing the new service level agreement using the automated negotiation.
15. The method of any of claims 10 through 14 wherein the QoS metric comprises one of computing capability or response time.
16. A processing device operable in a computing service, comprising:
a non-transitory memory storage comprising instructions; and
one or more processors in communication with the memory storage, wherein the one or more processors execute the instructions to:

receive customer-preference data that identifies for a customer a relationship between consumption of one or more first resource types and one or more second resource types, the customer-preference data constrained by a service level agreement between the customer and the computing service;

receive a request for execution of an application from the customer by the computing service; and

manage the service level agreement and cloud service resources for the customer through a set of MPI and PMI processes, the processes including:

determining that the request does not violate that customer's service level agreement terms, and that the service level agreement service levels are maintained;

determining cloud service resource configurations for the at least one customer;

configuring a set of computing service resources to implement the resource configurations to execute the set of processes through a message passing interface and a processing management interface;

monitoring the set of processes based on a set of quality of service (QoS) metrics; and

based on the monitoring and the customer preference data, and a determination that one or more of the set of QoS metrics is not sufficient for the set of processes, automatically modifying the set of computing resources for the customer.

17. The device of claim 16 wherein the customer-preference data identifies a relationship for an application executed by the computing service on behalf of the customer.

18. The device of any of claims 16 through 17 wherein the customer-preference data identifies, for the customer, the relationship as a rule, the rule based on a metric defining conditions and rules for a measurement of computing service performance.

19. The device of any of claims 16 through 18 wherein at least one of the one or more first resource types comprises processor allocation.
20. The device of any of claims 16 through 19 wherein the resource is thread, core or socket allocation.
21. The device of any of claims 16 through 20 wherein the one or more processors execute the instructions to:
- initiate an automated negotiation with the computing service;
 - request a new service level agreement from the computing service; and
 - establish a new service level agreement using the automated negotiation.
22. The device of any of claims 16 through 21 wherein the QoS metric comprises one of computing capability or response time.
23. The device of any of claims 16 through 22 wherein the resource comprises one of computing nodes, cores or threads.
24. A non-transitory computer-readable medium storing computer instructions for one or more processors executed by a cloud computing provider, that when executed by one or more processors, cause the one or more processors to:
- receive, by the cloud computing provider, customer-preference data that identifies, for an application to be executed by the cloud computing provider for at least one customer, a relationship between consumption of a first resource type and a second resource type;
 - receive a service request to execute a set of processes on cloud computing provider processing resources for the customer; and
 - manage a service level agreement and resources for the customer through a set of processes via message passing interface and a processing management interface, including:

determining that the request does not violate that customer's service level agreement terms, and that the service level agreement service levels are maintained;

determining cloud computing service processing resource configurations for the at least one customer;

configuring a set of cloud computing resources to implement the resource configurations to execute the set of processes through a message passing interface and a processing management interface, the processors identifying each processing node, processing core, processing thread with a unique identifier;

monitoring the set of processes based on a set of QoS metrics; and

based on the monitoring and the customer preference data, and a determination that one or more of the set of QoS metrics is not sufficient for the set of processes, automatically modifying the set of computing resources for the customer.

25. The non-transitory computer-readable medium of claim 24 wherein the customer-preference data identifies, for a customer entity, the relationship as a rule, the rule based on a metric defining conditions and rules for a measurement of cloud computing provider performance.

26. The non-transitory computer-readable medium of any of claims 24 through 25 wherein at least one of the one or more first resource types comprises processor allocation.

27. The non-transitory computer-readable medium of any of claims 24 through 26 wherein the resource comprises one of sockets, cores or threads.

28. The non-transitory computer-readable medium of any of claims 24 through 27 wherein the one or more processors to further perform the steps of:

receiving a request to begin an automated negotiation with the cloud computing service;
negotiating a new service level agreement using the automated negotiation;
generating the new service level agreement from the cloud computing service;
and
establishing the new service level agreement using the automated negotiation.

29. The non-transitory computer-readable medium of any of claims 24-28 through 18 wherein the metric comprises one of computing capability or response time.

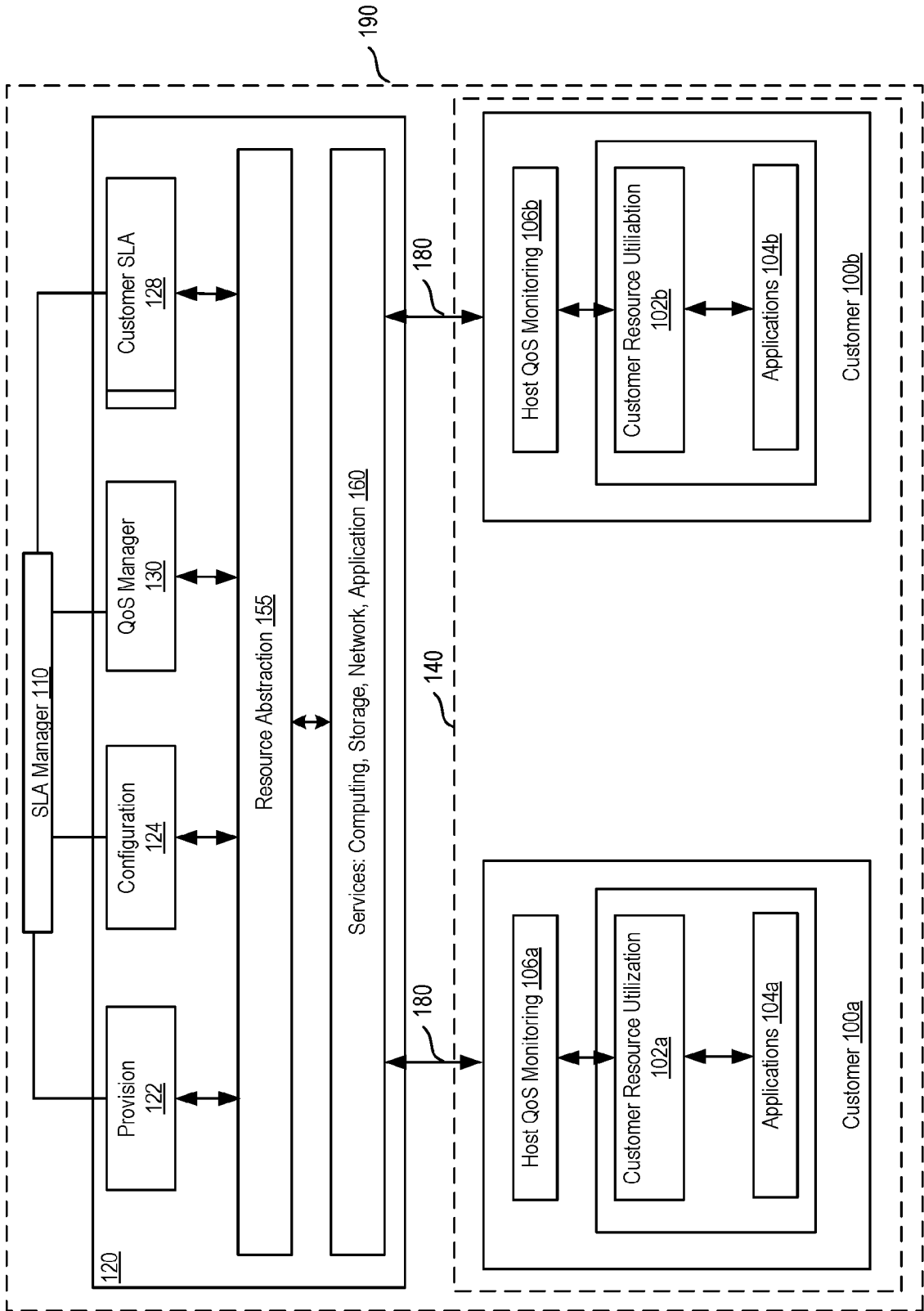


FIG. 1

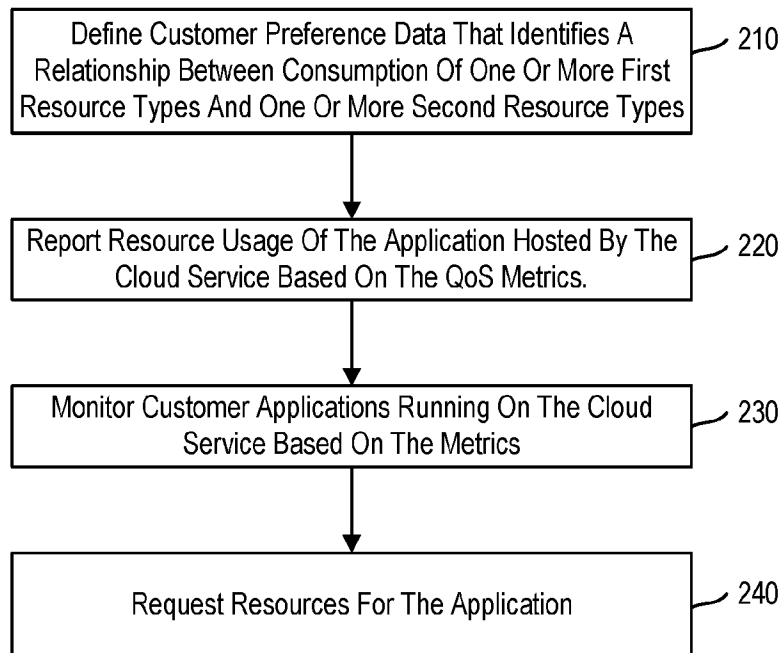


FIG. 2A

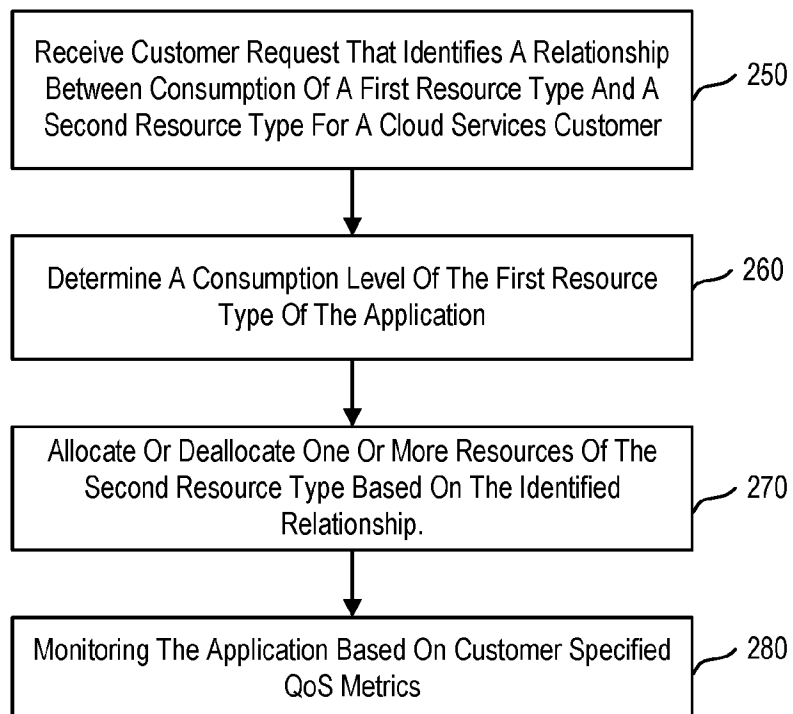


FIG. 2B

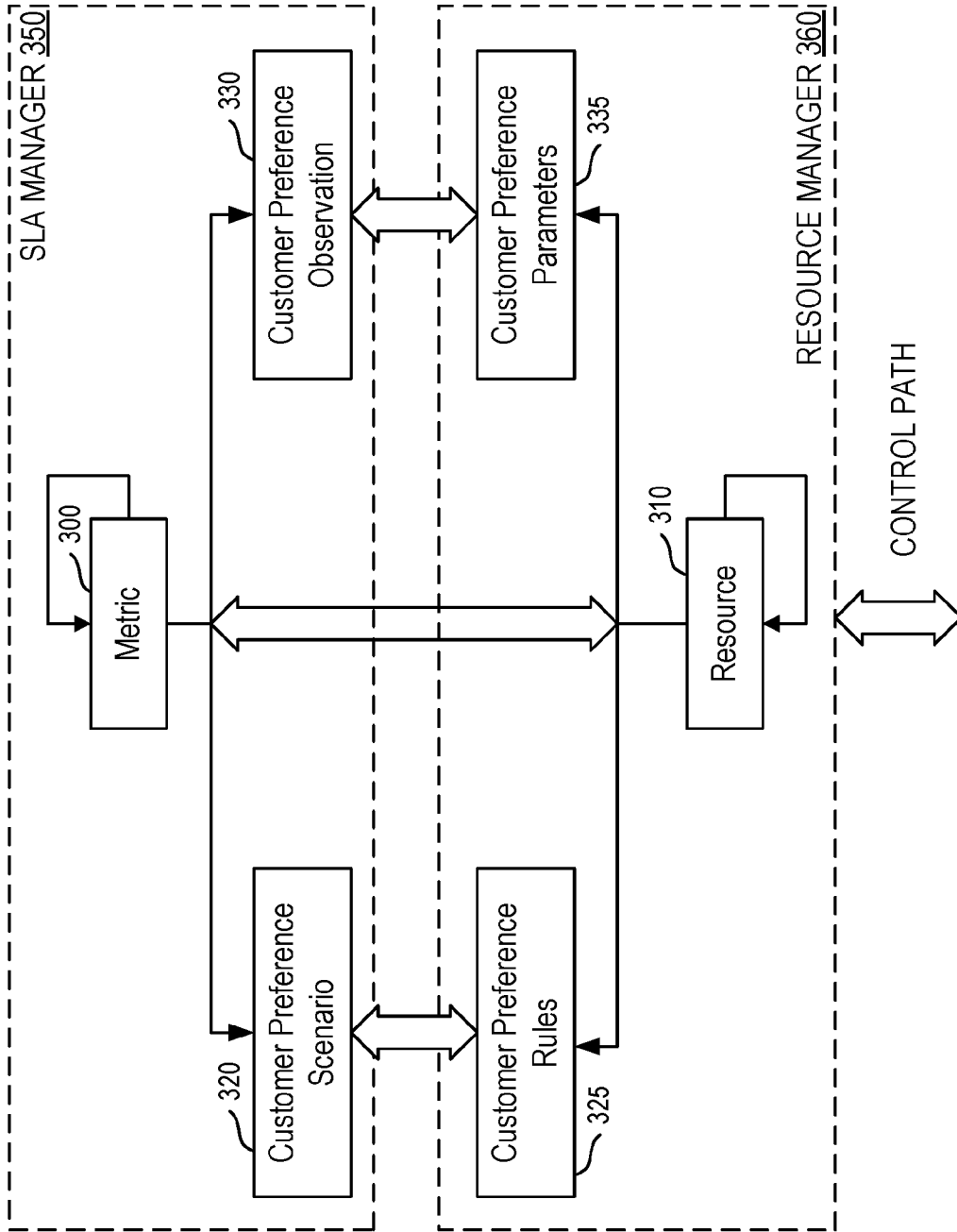


FIG. 3

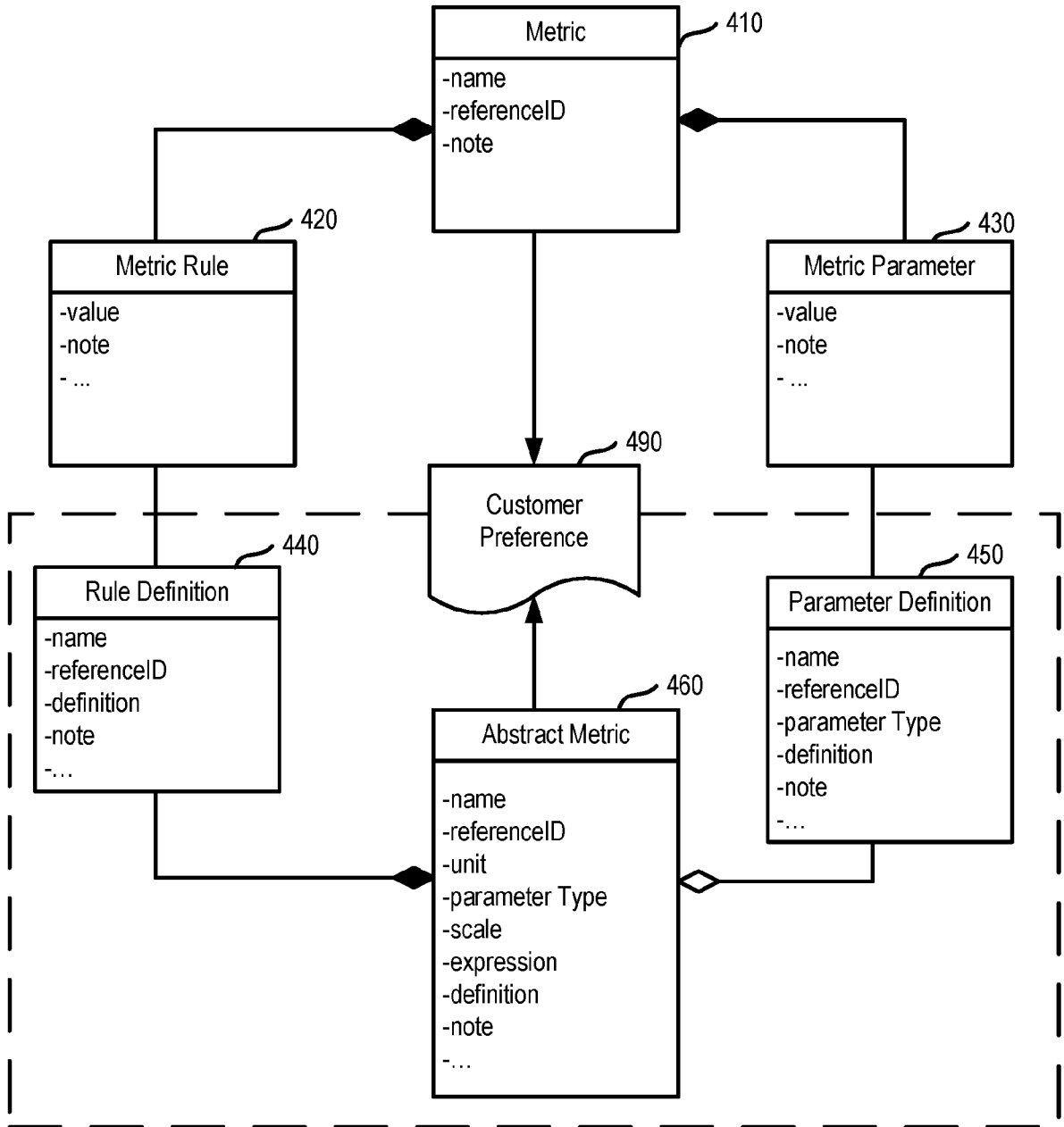


FIG. 4

Customer Preference Table		
Entity ID	Metric ID	Entity Type
1234	1	Application
2345	2	Organization
3456	3	Group

FIG. 5A

Metric Table			
Metric ID	Rule ID	Resource 1	Resource 2
1	11	101	102
2	12	101	103
3	13	104	Null

FIG. 5B

Rule Table	
Rule ID	Rule
11	If additional resources of RESOURCE 1 are allocated, also increase allocation of RESOURCE 2
12	If resources of RESOURCE 1 are deallocated, also decrease allocation of RESOURCE 2
13	If QoS drops below threshold, increase RESOURCE 1

FIG. 5C

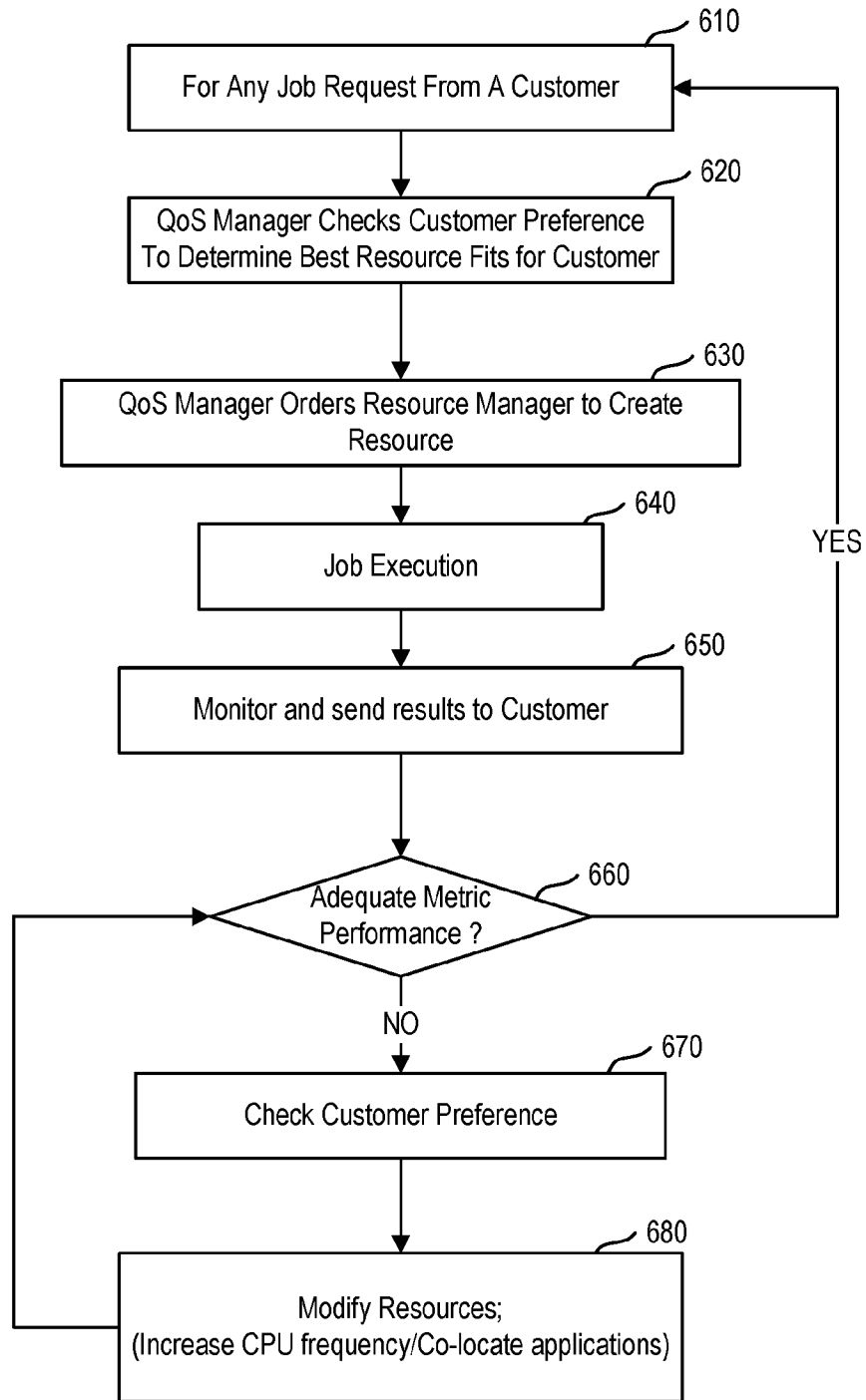


FIG. 6

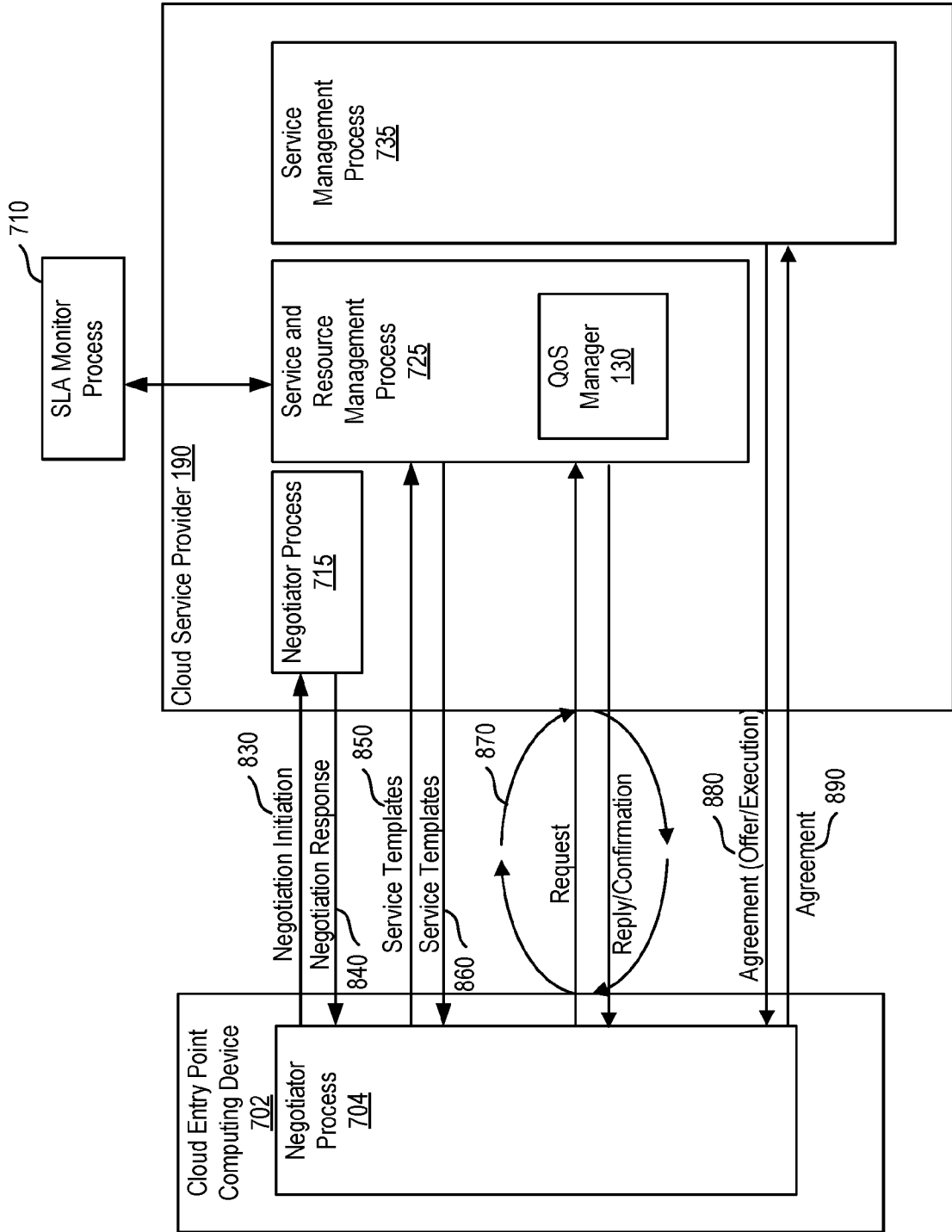


FIG. 7

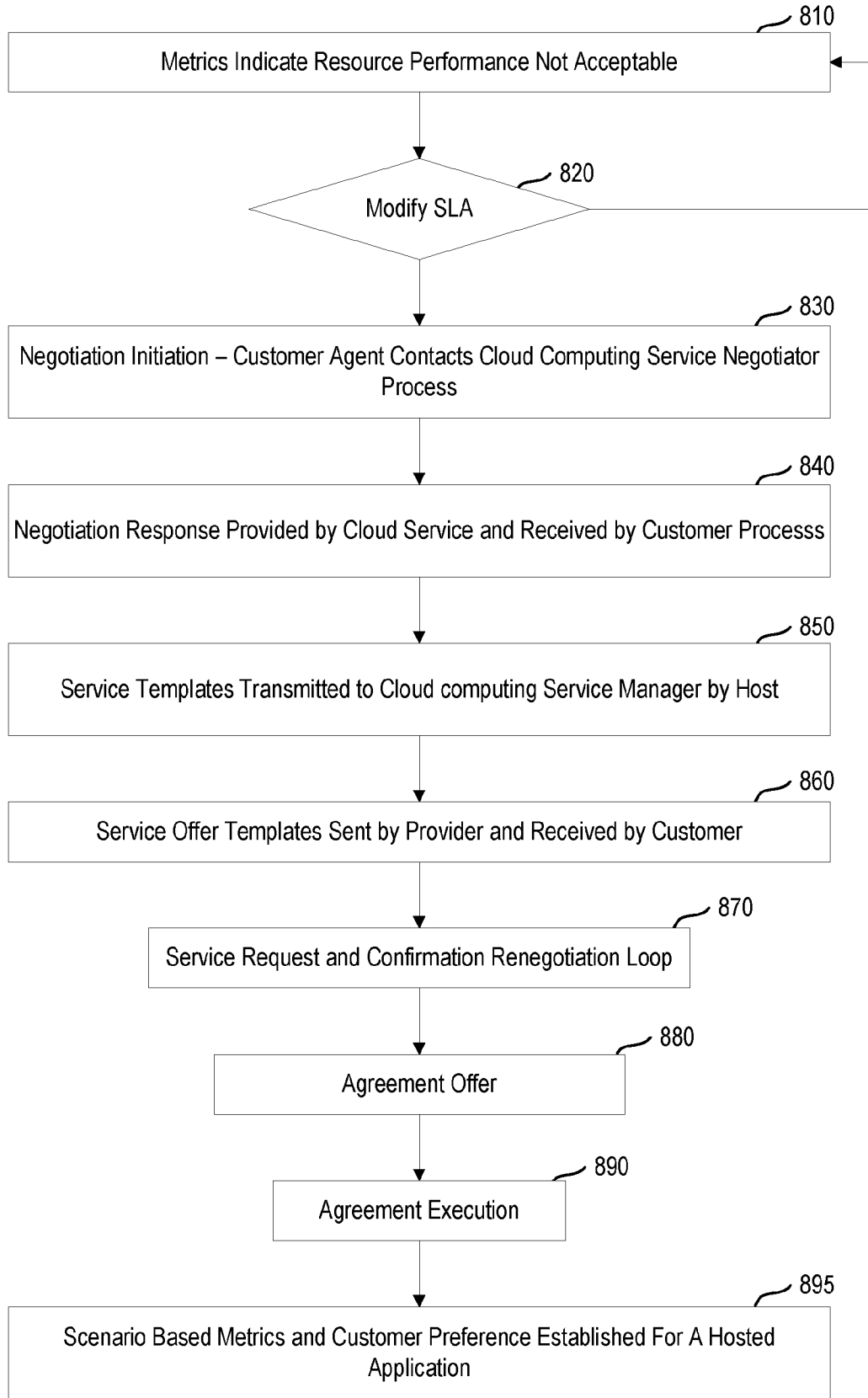


FIG. 8

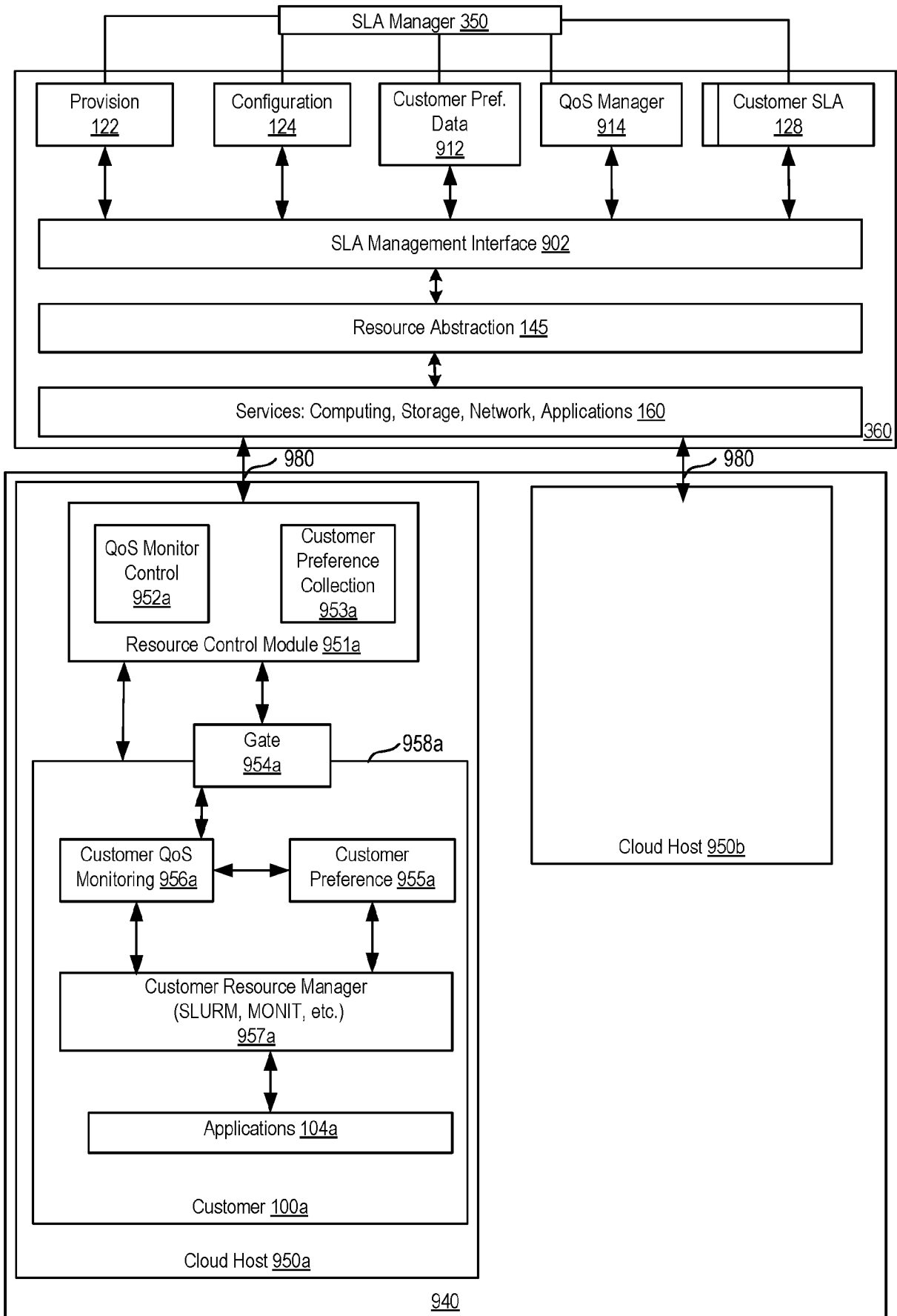


FIG. 9

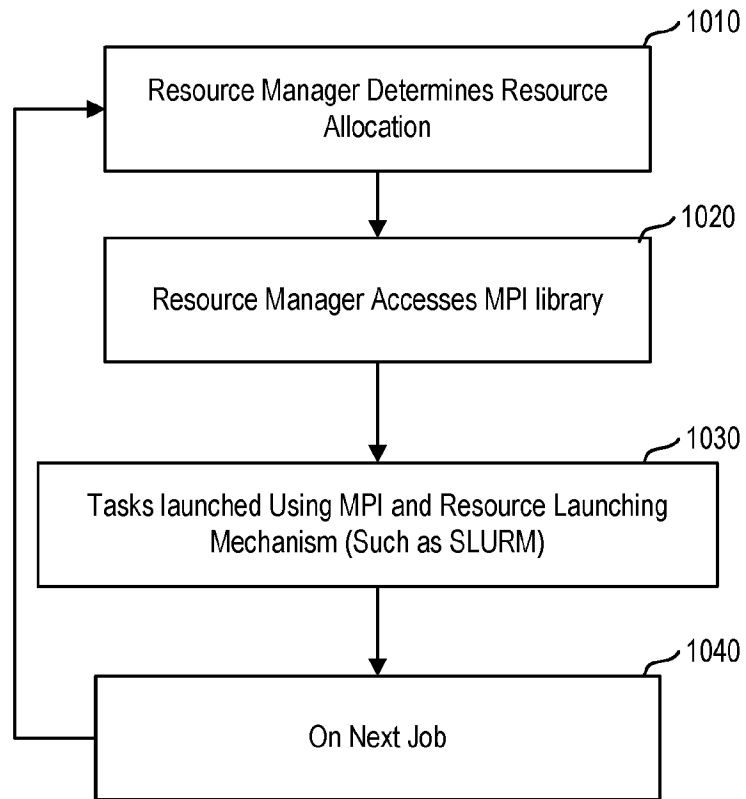


FIG. 10

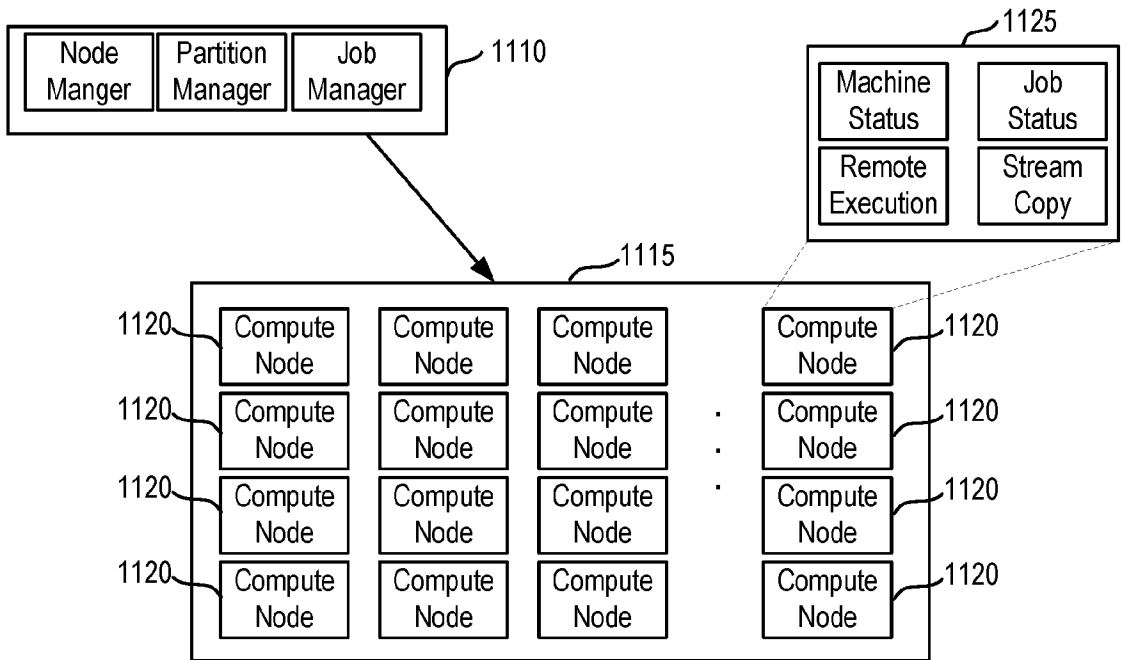


FIG. 11A

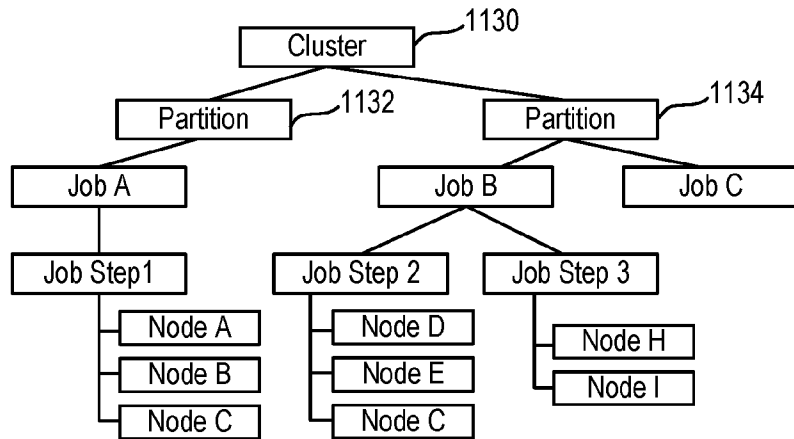


FIG. 11B

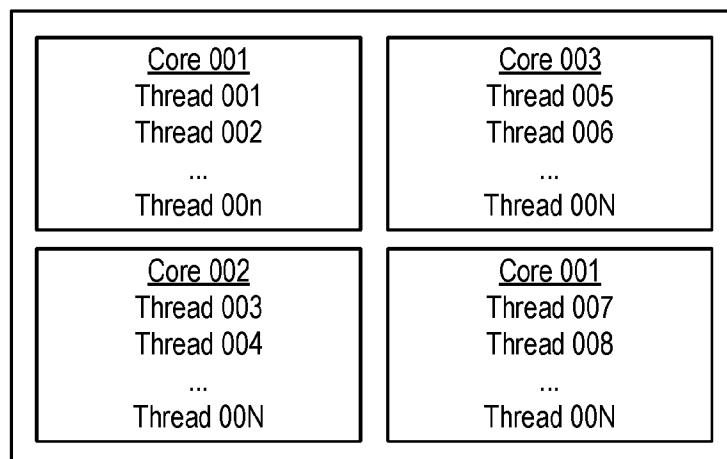


FIG. 11C

NodeName	N0	N1	N2	N3
Number of Sockets	2	2	2	2
Number of Cores per Socket	4	4	4	4
Total Number of Cores	8	8	8	8
Number of Threads Per Core	1	1	1	2
Number of CPUs	8	8	8	16

FIG. 12

PartitionName	Regnotes	Hypernote
Nodes	n0 n1 n3	n3
Default	YES	-

FIG. 13

Node Name		n0							
Socket ID		0				1			
# Allocated CPUs		3				3			
Allocated CPU IDs		0 1 2				4 5 6			
Binding of Tasks to CPUs	CPUID	0	1	2	3	4	5	6	7
	Task ID	0	2	4	--	1	3	5	--

FIG. 14

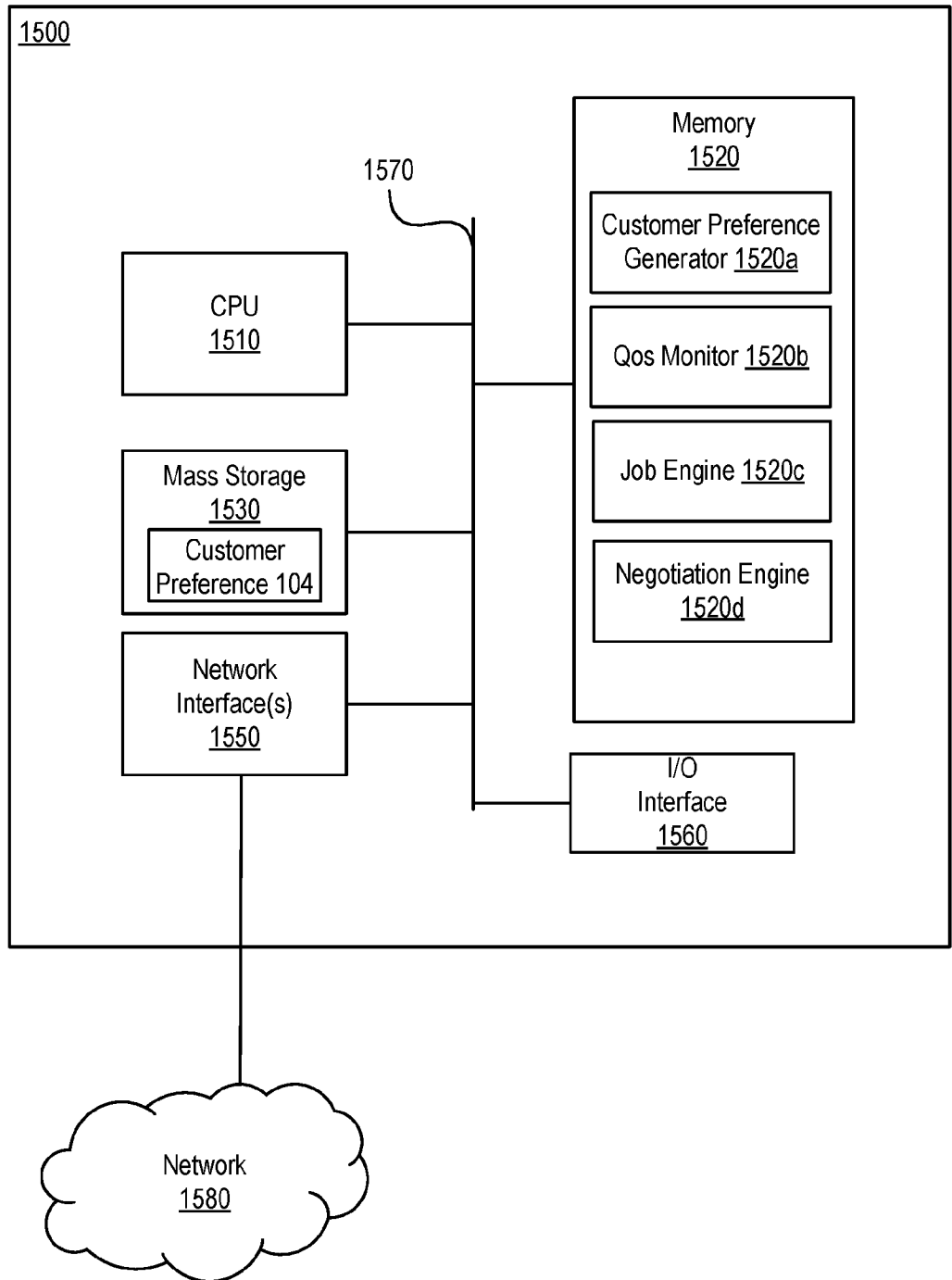


FIG. 15