



- (51) **International Patent Classification:**
G06Q 30/06 (2012.01) *G06Q 50/10* (2012.01)
G06Q 50/30 (2012.01)
- (21) **International Application Number:**
PCT/US2015/034431
- (22) **International Filing Date:**
5 June 2015 (05.06.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** BALESTRIERI, Filippo; 1501 Page Mill Road, Palo Alto, California 94304 (US). DREW, Julie Ward; 1501 Page Mill Road, Palo Alto, California 94304 (US). HUBERMAN, Bernardo; 1501 Page Mill Road, Palo Alto, California 94304 (US).
- (74) **Agents:** PAGAR, Preetam B. et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (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, 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, 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,

[Continued on next page]

(54) **Title:** PRICING OF CLOUD RESOURCES

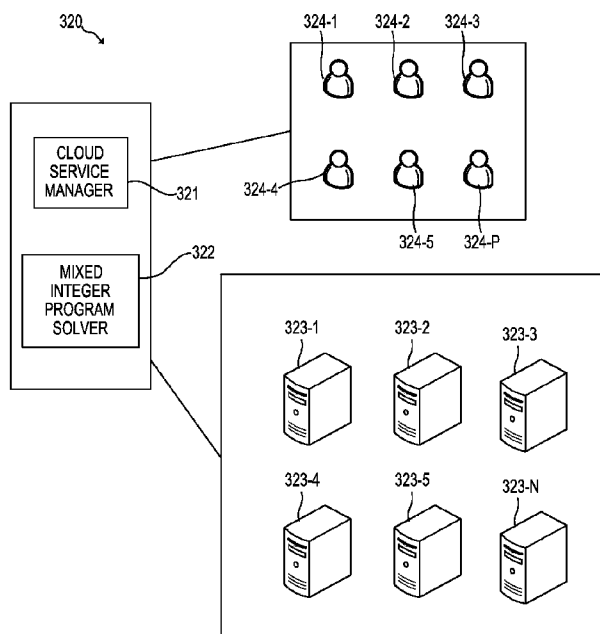


Fig. 3

(57) **Abstract:** Techniques are provided for pricing of cloud resources. A system for pricing of cloud resources can include a request engine capable of continuously receiving requests for cloud services from a plurality of users of a cloud system and an output engine to determine a plurality of variables relating to each of the continuously requested cloud services using a plurality of parameters received. The system can include a menu engine to generate a cloud service menu to be offered to the users in response to a subset of the received requests to be offered to the plurality of users, the cloud service menu defining a price per-resource-period for each of the requested cloud services based on the determined variables and a resource allocation engine to allocate cloud resources to fulfill the requested cloud service in response to a user selection from the cloud service menu.



SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

— as to applicant's entitlement to apply for and be granted
a patent (Rule 4.17(ii))

Declarations under Rule 4.17:

— as to the identity of the inventor (Rule 4.17(i))

Published:

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

PRICING OF CLOUD RESOURCES

Background

[0001] The Cloud refers to hardware and software resources available across the Internet. Companies that offer these resources are referred to as Cloud Service Providers (CSP). Cloud services can be roughly categorized as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). This categorization is based on the complexity of the service, from raw compute resources, such as storage or processing power, to refined software services, such as databases or other applications.

Brief Description of the Drawings

[0002] Figure 1 illustrates an example system for pricing of cloud resources according to the present disclosure.

[0003] Figure 2 illustrates a diagram of an example computing device according to the present disclosure.

[0004] Figure 3 illustrates a diagram of an example system for pricing of cloud resources according to the present disclosure.

[0005] Figures 4A-4D illustrate example menus for price, completion time, and resource allocation determination for cloud services according to the present disclosure.

[0006] Figure 5 illustrates an example flow chart of a method for pricing of cloud resources according to the present disclosure.

Detailed Description

[0007] The cloud computing model allows users (e.g., customers) to rent computing infrastructure as needed, rather than having to purchase their own resources. Moreover, customers can increase or decrease the size of their cloud resources in a straightforward and timely manner, depending on their computing conditions and/or cloud services needed by the user. Cloud services may be sold according to an instance-based model. In an instance-based model, the customer may rent the instances needed to complete a particular job. As used herein, an “instance” refers generally to an isolated user space instance, which can be executed within a virtualized environment (e.g., a “cloud”). Other technologies aside from hardware virtualization can provide isolated user space instances, also referred to as data compute nodes. Data compute nodes may include non-virtualized physical hosts, virtual machines (VMs), containers that run on top of a host operating system without a hypervisor or separate operating system, and/or hypervisor kernel network interface modules, among others.

[0008] A batch application, herein referred to as a “service” a “job”, or a “cloud service”, may include the usage of resources such as virtual servers or nodes. A cloud service may be a deployable unit comprised of one or more job steps. An example of a cloud service may be a Monte Carlo simulation. Another example of a cloud service may be an optimization problem. Another example is a data mining job. Put another way, a cloud service may be a single well defined job that includes a plurality of processes to be completed. A cloud service may include a series of closely related processing steps that, taken together, perform a particular process and/or processes.

[0009] Cloud services may be sold according to a resource-based model in which the customer rents the resources (nodes or instances per unit time) from the cloud service provider. The cloud services may be sold according to various resource-based selling models: on-demand instances, reserved instances, and

spot instances. On-demand instances allow users to rent nodes (when nodes are available) at a fixed hourly rate without any long term commitment. Some providers offer an additional feature with on-demand instances, where users can set a maximum budget on their cloud expenses, thus enabling the users to control costs. Reserved instances are nodes that are also rented at a fixed rate per unit time but may include a long-term commitment (e.g., one or more years) and upfront payment from the user. In exchange for this long term commitment the user may receive a lower hourly rate and a guarantee (subject to service level agreements) that the nodes will be available when needed. Further, some providers maintain a market through which users can resell their unused reserved instances.

[0010] Some providers also offer a resource-based selling model, referred to as spot instances, with dynamic spot prices expressed in dollars per hour. In this market, users may bid for spot instances by providing a maximum price they are willing to pay per instance per hour. The instances supplied are the ones that have not been sold as on-demand or reserved instances. At each point of time the spot price is the price at which the market clears (e.g., demand matches supply). If a user's maximum price bid exceeds the current spot price, his request is fulfilled and his instances will run until either he chooses to terminate them or the spot price increases above his maximum price (whichever happens sooner).

[0011] Pricing of cloud resource in accordance with the present disclosure allows a cloud service provider to allocate nodes to different services and provide pricing options to users corresponding to the amount of node-periods employed during a period of time to complete the service. As such, the user does not need to rent more node-periods than are needed for their job. Yet unlike the on-demand instances model, the user may make a commitment for a particular number of node-periods that may be required by the job and may receive corresponding price reductions from doing so. This commitment may allow the provider to make more efficient use of his resources. In this case the commitment is only for the conditions of the job, with typically a far shorter duration than the year or longer commitment

associated with reserved instances. Also, unlike the spot market model, the prices charged for the node-periods are set by the provider.

[0012] Further, examples of the present disclosure, unlike other approaches, considered customers' willingness to pay as a function of a completion time. Additionally, examples of the present disclosure are "online" cases, meaning information may be continuously received by the provider. Put another way, examples of the present disclosure have the capability of receiving information continuously; however, in some examples, information can arrive at any time, at any frequency, etc. As used herein, continuously received means information comes in over time. "Continuously receiving" may also be referred to as "capable of continuously receiving."

[0013] In some examples, rather than assuming all job requests are known upfront, it is recognized that job requests may reveal themselves over time. The online nature of the environment gives the provider recourse to improve upon earlier resource allocation decisions based on information that is revealed over time. For instance, the provider can receive requests and update dynamically, meaning the cloud service receive requests in a usually continuous, productive, or changing manner and updates in the same manner. In some examples, because of the online nature, prices and durations of requests arriving may be determined before future requests are known.

[0014] Some other approaches include examples where customers do not buy instances, but rather they describe the job they want, set a number of potential completion times, and provide their willingness to pay with respect to each completion date. Given such inputs, the cloud provider evaluates the benefit of accepting the job (through an automatic mechanism) and comes back with a response (positive or negative) for the customer. In contrast, examples of the present disclosure include the customers' willingness to pay as a function of completion time remaining their private information, whereas the other approaches assume that the customer reveals this information. Further, some other approaches consider the sale of cloud services based on completion times, in

contrast to the present disclosure which includes a model based on resource usage. Also, some other approaches consider an environment in which all requests and associated willingness to pay information are known upfront, whereas the present disclosure considers an online environment where requests become known over time. Examples of the present disclosure also consider an uncertainty dimension over the customers' willingness to pay and over services' total resource cost.

[0015] In still other approaches, focus is placed on how the allocation of resources is affected when the service completion time (urgency) is taken into account for a given set of services. These approaches bypass the selling stage of the interaction between service provider and potential customers, and, therefore, they refrain from considerations regarding how to extract surplus from customers interested in different service completion times. In contrast, examples of the present disclosure combine pricing and resource allocation to implement resource-based sales of cloud services in an online environment with completion-time sensitive customers.

[0016] Moreover, a cloud service provider may have limited resources. The cloud service provider may face a set of potential customers requesting different services. Each service may be described in terms of arrival time (also referred to as earliest start time) and total resource cost. The total resource cost, as used herein, refers to the number of resources that are needed to complete the service (e.g., in node-periods). It may depend on the application and the data that constitutes the service. The number of resources assigned over time to a job affects the speed at which the job is completed. The service provider may choose how to assign resources to the service over time in order to satisfy the total resource cost condition. Furthermore, each potential customer may be willing to pay different amounts of money if the service requested is completed within different time intervals (e.g. 1-3 hours, 1-12 hours...).

[0017] Still further, pricing of cloud resources may be subject to three or more sources of uncertainty: uncertainty about the sequence of requests that will

arrive over time; uncertainty over the customers' willingness to pay for various cloud services; and uncertainty over the total resource cost associated with each service. The service provider may desire to profit by selling a menu of different prices per node-period. The service provider may decide which items to include in the menu, how to price each item in the menu, and how to allocate resources in order to deliver the services sold according to the agreed per-resource price, among other considerations.

[0018] Pricing of cloud resources in accordance with the present disclosure provides a model by which a cloud provider may sell cloud services in a resource-based system. Examples of the present disclosure may include pricing and scheduling resource-based cloud services to increase a service provider's revenue in a market of cloud batch-applications in which customers' willingness to pay depends on job completion times. For instance, examples include determining how to price resources over time, and how to execute services to increase (e.g., maximize) the provider's revenue. Other examples of the present disclosure can maximize profits for the provider.

[0019] Examples may produce a per-resource-period price that depends on the time in which a request for nodes is submitted and/or on the duration in which the request is completed. As used herein, a per-resource-period price refers to a price for a particular cloud service that is based on the number and type of resources reserved for the cloud service for a particular period of time. A customer may determine whether to accept the price offered given his willingness to pay, which is his private information. Given each customer's choice, examples may allocate resources to sold services. In some examples, a service provider may receive service requests over time (e.g., continuously, dynamically), and can maximize a profit by assigning resource prices in each time period to requests arriving in that period. Examples of the present disclosure may include systems, methods, and machine-readable and executable instructions and/or logic.

[0020] Pricing of cloud resources according to the present disclosure allows a practical implementation of resource-based sale of cloud services. While the

foregoing disclosure generally refers to a “customer” and a “user” interchangeably, it is noted that a “customer” may be either an internal customer or an external customer. Put another way, the cloud services may be offered to users within the same company and/or organization as the cloud service provider. In such examples, the objective function (described further herein) may be to increase benefits of the cloud services derived by the users within the company and/or organization. For instance, examples of the present disclosure may be applied to distribute private cloud resources across different departments within an enterprise. In that case, a per-resource-period price is the transfer price that a department is charged for using enterprise resources, and the objective is to maximize the social surplus/efficiency within the company (i.e. the aggregate willingness to pay of company departments that submit requests).

[0021] Interaction between each customer and the cloud resource provider in any period t^* can be described as follows. In an example, at time t^* , a number of requests j arrive. These are requests for which $s(j)=t^*$. The cloud provider sets the menu of per-node-period prices for the jobs arriving t^* , indexed by duration: $\{p(t^*,d)\}$. Given the menu $\{p(t^*,d)\}$ and the customer's willingness to pay $w(j,k,d)$, the customer submitting request j with arrival time t^* may decide whether to buy $r(j,h)$ nodes or not and, if the customer buys, the customer selects the completion duration d (by selecting one price $p(t^*,d)$ from the menu). In this example, the quantity bought by the customer is either $r(j,h)$ or 0. If the customer makes a selection, the provider learns $r(j,h)$. Further, after the selection is made by the customer, the provider may decline to serve a request if the provider so chooses.

[0022] Figure 1 illustrates an example system 100 for pricing of cloud resources according to the present disclosure. The system 100 may include a database 101, a subsystem 102, and/or a number of engines (e.g., request engine 103, an output engine 104, a menu engine 105, and a resource allocation engine 106). As used herein, “a” or “a number of” something can refer to one or more such things. For example, “a number of widgets” can refer to one or more widgets. The subsystem may include the number of engines in communication with the

database 101 via a communication link. The system 100 may include additional or fewer engines than illustrated to perform the various functions described herein. The system may represent software and/or hardware of a network controller (e.g., device 208 as referenced in Figure 2, etc.).

[0023] The number of engines may include a combination of hardware and programming that is configured to perform a number of functions described herein (e.g., receive a request for a cloud service from a user of a cloud system, etc.). The programming may include program instructions (e.g., software, firmware, etc.) stored in a memory resource (e.g., computer readable medium (CRM), machine readable medium (MRM), etc.) as well as hard-wired program (e.g., logic). The request engine 103 may include a combination of hardware and programming that is configured to be capable of continuously receiving requests online from a plurality of users (e.g., customers) of a cloud system. As used herein, a cloud service request received online refers to a cloud service request that is submitted while other jobs are running/executing on the cloud service. For instance, requests from customers may arrive at the cloud service provider at any time (e.g., continuously, sequentially). Put another way, the requests are received dynamically. As used herein, continuously receiving (or capable of continuously receiving) requests refers to receiving requests by the cloud service provider while the cloud service provider is using resources. At each point in time, the cloud service provider receives different requests to execute jobs. As used herein, dynamically receiving requests refers to the cloud service receiving requests in a usually continuous, productive, or changing manner. In such instances, certain nodes in the cloud service may be in use, meaning an incoming request may not be fulfilled using that occupied node.

[0024] In the foregoing examples, "time" is considered a sequence of discrete intervals (e.g. minutes, hours, days, etc.). Given that, a set of time periods $T = \{1, \dots, T\}$ may be considered. In each period, a request for nodes may be submitted, and each request can start to be served as soon as it is submitted. Examples include online scheduling, meaning that, at the beginning of the first

period, only the service requests submitted in the first period are known. The provider may have some information about future requests, but this information may change over time. The provider may make pricing and resource allocation decisions for jobs submitted in the current period, without full knowledge of future requests. As time progresses, new requests may be submitted. As they are submitted, the provider learns about them and makes pricing and resource allocation decisions for the new requests, while honoring completion-time commitments and prices for prior requests. In some examples, the provider may change the allocation of resources for prior requests as long as he completes those requests in the committed time.

[0025] Examples of the present disclosure may be formulated with respect to a particular time period t^* representing the present time. The resource conditions (e.g., requirements) and the customers' decisions to accept or reject contracts may be known for requests submitted prior to period t^* . The provider may determine at time t^* target (e.g., optimal) prices and resource allocation decisions for requests arriving in in time period t^* , given the commitments the provider made for prior requests and updated information regarding current and future requests.

[0026] The last period T defines the finite horizon length over which the mechanism is applied. The horizon length may be determined by external factors (e.g. there is a general system reset every 24 hours) or may be selected arbitrarily by the service provider. The horizon length used in the model may differ by the period t^* in which it is solved. For example, in period $t^*=1$ the model may have time periods $\{1, \dots, 20\}$ and in $t^*=2$ it may extend the horizon by one period, using time periods $\{1, \dots, 21\}$. This practice of extending the horizon as time progresses, also known as rolling horizon planning, allows a provider to base the most immediate decisions (pricing of jobs arriving in t^*) on information in the short-term future and not on forecasts far into the future.

[0027] In some examples, the system 100 may include an optimization engine (not illustrated in Figure 1). The optimization engine may include a combination of hardware and programming that is configured to dynamically define

(e.g., optimize) the price per-resource-period as new requests for cloud services are received. As used herein, dynamic optimization refers to adjusting offers of prices per-resource as new information is obtained and/or learned. For instance, as new customer and job information is received in a dynamic (e.g., continuous, productive, changing, etc.) manner, offers may change to better maximize profits and/or efficiency.

[0028] An input engine (not illustrated in Figure 1) may include a combination of hardware and programming that is configured to receive parameters relating to the requested cloud service from at least one of the user and a provider of the cloud system. Put another way, the user (e.g., customer) may provide parameters, the cloud service provider may provide parameters, or both the user and cloud service provider may provide parameters.

[0029] As used herein, a parameter refers to an input that has a known value. An example of a parameter may be the earliest period in which the job, represented by j , can start, the parameter represented by $s(j)$. The value of $s(j)$ may correspond to the job j 's arrival time or to the earliest time the job j may begin. Another example of a parameter may be the total number of node-periods that the job j may require for completion if the job j is of a particular type (e.g., type h), the parameter represented by $r(j, h)$. Yet another example of a parameter may be (an estimate of) of a customers' willingness-to-pay level for the job j if the customer requesting job j is of a particular type (e.g., type k), the parameter represented by $w(j, k, d)$. Put another way, the cloud service provider may make particular assumptions about the customer k 's willingness to pay for job j completion in duration d . Other examples of parameters include: $\mu(j, h)$, the probability that job j is of type h , and $\pi(j, k)$, the probability that job j 's customer is of type k .

[0030] In some examples, parameters j , $s(j)$, and $r(j, h)$ come from the user, while other variables are provided by the cloud provider. Embodiments are not so limited, however. Additionally, in some examples, the timing according to which the parameters are entered may differ. Parameters provided (e.g., entered) by the

cloud provider are provided before any price $p(t^*, d)$ is offered. An engine may be pre-set up with these provider-provided parameters.

[0031] The output engine 104 may include a combination of hardware and programming that is configured to determine a plurality of variables relating to each of the continuously requested cloud services using a plurality of parameters. In some instances, a mixed integer programming solver may also be used. As used herein, a mixed integer programming solver refers to hardware and programming that is configured to solve an optimization problem in which some or all of the variables are discrete integers, and some of the variables are allowed to be non-integers or continuous. Examples of mixed integer programming solvers may include CPLEX® and/or Gurobi®. However, examples are not so limited, and the system 100 may be implemented with a different mixed integer programming solver than those described. Discrete integers may include binary variables 0 and 1, whereas non-integers may include all numerical values between two integers, for example, 0 and 5 (e.g., 0, 1, 2.5, 4.1, etc.).

[0032] A variable may refer to an endogenous variable that may be determined from the system 100. As discussed further in relation to Figure 3, examples of variables may include an integer number of node-periods, a maximum number of nodes, a binary variable indicating if a particular service request having a particular cost type and customer type is completed at a particular time, a binary variable indicating if a service request having a particular cost type and customer type is completed with a particular elapsed time, a revenue earned for a particular service request, and a price offered for a particular service request, among other variables. For example, based on the parameters received, either from the user or the cloud service provider, $x(j, h, k, t)$ may be determined, the integer number of node-periods assigned to job j of cost type h at time t for customer type k . Furthermore, based on the parameters received, $p(t, d)$, referred to as the price per-resource, may be determined. As used herein the price per-resource refers to the price offered per node-period for any request that arrives in period t and is completed in d periods. Thus, the price for a job j , if the cost type is h and if the job

is served (e.g., performed by the cloud service provider) and is completed in d periods, is $p(s(j), d) r(j, h)$. In such a manner, a plurality of pricing options may be determined for each job.

[0033] The price $p(t, d)$ depends on t and d but not j . As such, the same menu applies for all jobs arriving in the same period t . If multiple jobs arrive in the same period, they are offered the same menu. For instance, generating a cloud service menu in response to continuously received requests includes generating menus shared across jobs j sharing the same earliest start time $s(j)$.

[0034] In some examples, the user and the cloud service provider may be entities within a same organization, such as within a private cloud or enterprise system. In such examples, the per-resource-period price for a requested cloud service may be a transfer price that the user is charged for the requested cloud service within the organization. For instance, if the user is a marketing department within an enterprise, the per-resource-period price may be a “cost” or transfer price that the marketing department is “debited” or “charged” by the company IT department for utilizing the cloud services associated with the requested cloud services.

[0035] The menu engine 105 may include a combination of hardware and programming that is configured to generate a cloud service menu to be offered to the users in response to a subset of the received requests to be offered to the plurality of users, the cloud service menu defining a price per-resource-period for each of the requested cloud services based on the determined variables. Based on the values of the variables determined, various menu options may be provided to a user. An example menu is discussed further herein with respect to Figures 4A-4D. For instance, the menu engine 105 may generate a menu having three prices listed to complete a requested cloud service: a first price (e.g., lowest) that has a first duration of time to complete the request, a second price (e.g., intermediate) that has a second duration of time to complete the request, and a third price (e.g., the highest) that has a third duration of time to complete the request. Examples are not so limited, however, and the cloud service menu may

include more or fewer pricing and reservation options than described herein. For instance, the cloud service provider may offer no menu at all to some cloud service requests, or only one (1) menu item to some cloud service requests. Put another way, the cloud service menu may differ from one cloud service request to the next.

[0036] In some examples, the system 100 may include a selection engine (not illustrated in Figure 1). The selection engine may include a combination of hardware and programming that is configured to receive and/or record a user's selection of a particular menu item.

[0037] The resource allocation engine 106 may include a combination of hardware and programming that is configured to allocate cloud resources to fulfill the requested cloud service in response to a user selection from the cloud service menu. In some examples, the resource allocation engine 106 may determine whether to accept the user selection based on available resources. For instance, the cloud service provider may determine that insufficient resources are available to reserve nodes for a request for a particular duration of time, in which instance the resource allocation engine 106 may reject the user selection from the cloud service menu. Further, the resource allocation engine 106 may determine that a user has declined the offers in the cloud menu after a threshold period of time has elapsed and a user selection has not been received. In a number of examples, the resource allocation engine 107 may execute jobs according to the selected menu item. For example, if a user among the plurality of users of the cloud system purchased a particular pricing option for a requested cloud service, resources in the cloud system may be allocated to specifically perform that requested cloud service for the agreed upon period of time and for the agreed upon price reflected in the cloud menu. In a number of examples, execution of a job is a fluid operation that may change (in terms of node allocation) as new requests are received.

[0038] Figure 2 illustrates a diagram of an example computing device 208 according to the present disclosure. The computing device 208 may utilize software, hardware, firmware, and/or logic to perform a number of functions described herein. The computing device 208 may be any combination of hardware

and program instructions configured to share information. The hardware, for example, may include a processing resource 209 and/or a memory resource 211 (e.g., CRM, MRM, database, etc.). A processing resource 209, as used herein, may include any number of processors capable of executing instructions stored by a memory resource 211. Processing resource 209 may be implemented in a single device or distributed across multiple devices. The program instructions (e.g., computer readable instructions (CRI)) may include instructions stored on the memory resource 211 and executable by the processing resource 209 to implement a desired function (e.g., receive a request for a cloud service from a user of a cloud system). While Figure 2 generally illustrates a single computing device 208, examples are not so limited. For instance, the system 100 (described in Figure 1) may be implemented on a plurality of computing devices, such that each engine described in Figure 1 and each module described in Figure 2 may be executed by one or more computing devices.

[0039] The memory resource 211 may be in communication with a processing resource 209. A memory resource 211, as used herein, may include any number of memory components capable of storing instructions that may be executed by processing resource 209. Such memory resource 211 may be a non-transitory CRM or MRM. Memory resource 211 may be integrated in a single device or distributed across multiple devices. Further, memory resource 211 may be fully or partially integrated in the same device as processing resource 209 or it may be separate but accessible to that device and processing resource 209. Thus, it is noted that the computing device 208 may be implemented on a participant device, on a server device, on a collection of server devices, and/or a combination of the user device and the server device. As used herein, servers may also be referred to as “nodes”.

[0040] The memory resource 211 may be in communication with the processing resource 209 via a communication link (e.g., a path) 210. The communication link 210 may be local or remote to a machine (e.g., a computing device) associated with the processing resource 209. Examples of a local

communication link 210 may include an electronic bus internal to a machine (e.g., a computing device) where the memory resource 211 is one of volatile, non-volatile, fixed, and/or removable storage medium in communication with the processing resource 209 via the electronic bus.

[0041] A number of modules (e.g., request module 213, output module 215, menu module 216, and resource allocation module 218) may include CRI that when executed by the processing resource 209 may perform a number of functions. The number of modules may be sub-modules of other modules. For example, the request module, the output module, the menu module, and the resource allocation module can be sub-modules and/or contained within the same computing device. In another example, the number of modules can comprise individual modules at separate and distinct locations (e.g., CRM, etc.).

[0042] Each of the number of modules may include instructions that when executed by the processing resource 209 may function as a corresponding engine as described herein. For example, the request module 213 may include instructions that when executed by the processing resource 209 may function as the request engine 103. Additionally, the output module 215 may include instructions that when executed by the processing resource 209 may function as the output engine 104.

[0043] The request module 213 may include instructions that when executed by the processing resource 209, may continuously receive from a plurality of users of a cloud system, a plurality of cloud service requests. As described in relation to Figure 1, cloud service requests may come in continuously, and may be received from a single user of the cloud system or from a plurality of users of the cloud system.

[0044] In some examples, an input module (not illustrated in Figure 2) may include instructions that when executed by the processing resource 209, may receive a plurality of parameters relating to the plurality cloud service requests from at least one of the user and a provider of the cloud system. As previously noted, some parameters may be provided by the user, while others are provided by the

cloud provider. Similarly, the output module 215 may include instructions that when executed by the processing resource 209, may determine a plurality of variables relating to the requested cloud service using the plurality of parameters and a mixed integer programming solver, as discussed in relation to Figure 1.

[0045] The computing device 208 may include a menu module 216 that when executed by the processing resource 209 may provide a cloud menu to a subset of the plurality of users for completion of a subset of the plurality of cloud service requests, the cloud menu including a price-per-resource-period for each of the subset of cloud service requests determined using the plurality of parameters and information gathered from previously provided price-per-resource-period options.

[0046] The menu module 216 may include instructions that when executed by the processing resource 209 may provide a cloud menu to a subset of the plurality of users for completion of a subset of the plurality of cloud service requests, the cloud menu including a price-per-resource-period for each of the subset of cloud service requests determined using the plurality of parameters. In some examples, the menu module 216 may provide a cloud menu to a subset of the plurality of users for completion of a subset of the plurality of cloud service requests. For instance, four cloud service requests may be received and cloud menus may be provided for only three of the received cloud service requests. In another example, no cloud menus may be provided. For instance, three cloud service requests may be received but no cloud menus are presented to the users.

[0047] Furthermore, the resource allocation module 218 may include instructions that when executed by the processing resource 209 may allocate cloud resources to fulfill a user-selected price-per-resource-period and duration option from the cloud menu. The duration may be implicit in the price $p(t,d)$.

[0048] In some examples, the computing device 208 may further include instructions that when executed by the processing resource 209 configured to process information regarding a job type (or a probability that a cloud service request among the plurality of cloud service requests is of a particular job type) for

each cloud service request received from users of the cloud system, and process information regarding a total number of node periods to complete each of the cloud service requests based on the determined job type and size. For example, a Monte Carlo simulation may include a different number of node periods as compared to an optimization problem. As discussed in relation to Figure 3, the computing device 208 may include instructions to process information regarding a probability that a cloud service request is received from a particular customer type. Based on the determined information, such as customer type, job type, cost type, etc., and using a plurality of constraints and/or rules, as discussed further, a price-per resource for execution of a requested cloud service may be determined to maximize the revenue received (by the cloud service provider) from the plurality of cloud service requests. The price-per resource, in some examples, may depend on at least one of a job arrival time and a service duration.

[0049] In some instances, the computing device 208 may further include instructions that when executed by the processing resource 209 configured to determine a revenue for each request if the request is of a particular job type, is for a particular customer type, and is completed in a specified time period.

[0050] Figure 3 illustrates a diagram of an example system 320 for pricing of cloud resources according to the present disclosure. As illustrated in Figure 3, the system 320 may include a cloud service manager 321, a mixed integer programming solver 322, and a plurality of cloud resources 323-1, ...323-N (collectively referred to as resources 323). As described herein, a cloud resource refers to a virtual or physical component in a cloud system. Examples of cloud resources may include VMs, virtual servers, physical servers, and/or nodes among other resources. Cloud resources may be exclusive, in that at each point of time, each resource can be applied only to one service. Furthermore, cloud resources may be re-usable in that past usage of a particular cloud resource does not affect future usability of the cloud resource. Additionally, cloud resources may be mobile in that a cloud resource that was utilized for a first service at one point of time may be reallocated to a second service at a second (e.g., subsequent) point of time.

Moreover, cloud resources may be non-perishable in that they do not expire after a certain time.

[0051] The cloud service manager 321 may deploy and manage cloud services. For instance, the cloud service manager 321 may receive requests online for cloud services from a plurality of users 324-1,..., 324-P (e.g., customers) of a cloud system. Put another way, requests for cloud services j may be collected continuously by the cloud service. Each service request may also be referred to as a “job”. Because of the continuous, online nature of the collection, a full set of jobs may not be known; rather sets of jobs develop dynamically. Each job may be indexed by j . Further, the set of jobs may be represented by $J = \{1, \dots, J\}$. Each job served may have a number of resources (e.g., nodes) assigned in possibly multiple time periods until completion. Before time period 1, the provider may have limited or no information about the sequence of request and their active times. Put another way, cloud service manager 321 may determine a plurality of resources 323 for each cloud service request.

[0052] The set of jobs $\{1, \dots, J\}$ may also differ by the time period t^* in which the model is solved. For example, the provider may learn new information about future jobs and incorporate this information in the set $\{1, \dots, J\}$. In the same way that the horizon may extend as time progresses, the provider may also extend the set of jobs to be considered.

[0053] As discussed previously herein, each job $j \in J$ may have the following characteristics:

$s(j)$: the arrival period of request j , and the earliest period in which it can be served.

$r(j, h)$: the total number of node-periods the job j may require for completion if the job j is of type h .

$\mu(j, h)$: the probability that job j is of cost type h . Note that $\sum_h \mu(j, h) = 1$ for each j .

[0054] Each service request (e.g., job) may be interpreted as a separate customer with a willingness to pay (schedule) for the particular job. The set of

customer types may be defined by $K = \{1, \dots, K\}$. Further, the service provider's belief of the customers' willingness to pay for the service may be represented by a probability distribution over different types of customers. For instance, different job and customer type combinations (j, k) may have the following characteristics:

$\pi(j, k)$: the probability that job j 's customer is of type k . Note that

$\sum_k \pi(j, k) = 1$ for each j .

$w(j, k, d)$: A customer of type k 's willingness to pay for job j if it is completed in d periods after arrival time $s(j)$.

[0055] As discussed previously herein, the total number of nodes or resources needed to complete a particular job may be determined. The total number of available nodes and/or resources for the cloud service provider may be represented by N and may be homogeneous.

[0056] Each job may have a horizon length. As used herein, the horizon length refers to the latest possible completion duration for job j that would result in revenue for the cloud provider. The horizon length may be determined using the following: $m(j) = \max_k \{\max\{d: w(j, k, d) > 0\}\}$. The inner maximization represents the maximum value for d for which $w(j, k, d)$ is positive. The outer maximization represents the maximum over k of the inner max value. In some examples, the horizon length may be represented by $T = \max_j \{s(j) + m(j)\}$ to ensure enough time in the horizon to complete every job received from the users 324. For each request, $l(j)$ may be the minimum number of periods to complete the request (if all N nodes are assigned to the request).

[0057] The cloud service manager 321 may receive a plurality of cloud service requests from users 324. Based on the cloud service requests received and expected to be submitted, the cloud service manager 321 may determine a plurality of resources 323 for each cloud service request received.

[0058] Based on the cloud service requests received and expected to be submitted from the users 324, the cloud service manager 321 may determine a plurality of variables (also known as endogenous variables and/or outputs) based

on job type and customer type. For instance, the following variables may be determined:

$x(j,h,k,t)$: integer number of nodes assigned to job j of cost type h at time t for customer type k . This may be defined for t such that $s(j) \leq t \leq s(j) + m(j) - 1$.

$g(j,t)$: maximum number of nodes allocated to job j in time t over all customer types k and cost types h . This may be defined for all j and for t such that $s(j) \leq t \leq s(j) + m(j) - 1$.

$y(j,h,k,t)$: binary variable indicating if job j of cost type h completes at time t for customer type k . This may be defined for t such that $s(j) + l(j) - 1 \leq t \leq s(j) + m(j) - 1$.

$u(j,h,k,d)$: binary variable indicating if job j of cost type h completes with elapsed time d from earliest start time for customer type k . This may be defined for d such that $l(j) \leq d \leq m(j)$.

$v(j,h,k,d)$: revenue earned for job j of cost type h during d periods after the job's arrival for customer type k . This may be defined for d such that $l(j) \leq d \leq m(j)$.

$p(t,d)$: price offered for per node-period used for any job that arrives in period t and is completed during d periods. Thus, the price for a job j , if the cost type is h and if the job j is served in duration d , is $p(s(j),d) r(j,h)$.

[0059] In some examples, a plurality of scenarios may be constructed in which the job may be completed. For each of those scenarios, a plurality of resources may be selected in order to complete the job. However, as described herein, the cloud service manager may also decide not to assign any resources to a particular cloud service request. In such examples, the cloud service manager may determine that a different job with a higher return on resource investment should be executed. Furthermore, the cloud service manager 321 may determine the total resource cost for each possible job. In other words, each job may be associated with a total resource cost. As used herein, a total resource cost refers to the total number of node-periods to complete a particular job. In a number of

examples, the total resource cost and the allocation of resources may be jointly determined to maximize the cloud service provider's revenue received from the plurality of cloud service requests.

[0060] The total resource cost for a request may be private information of the customer submitting the request. From the cloud provider perspective, it may be a probability distribution over different values. A degenerate probability distribution (probability 1 on a specific resource amount r) for request j may imply that its resource cost is known to the cloud service provider *ex ante*. This may be because it is common knowledge, such that both the customer and cloud service provider learn it once the actual job is submitted. In some examples, after the request has been submitted, the provider has offered a price menu, and the customer has selected a menu item, the resource cost for the request becomes known to the provider. To capture the provider's uncertainty over the requests' costs at the moment of setting the prices, different cost types h for each job j may be considered.

[0061] A request may be considered as work or a job that needs to be completed, but in some examples, it may be a node-period reservation by the customer. The customer may use the node-periods for any purpose they choose, including leaving them idle for the duration of the request's "processing" time. At time t^* , the provider may know the resource cost $r(j,h)$ for jobs j that have been submitted prior to time t^* , if the customer has accepted one of the provider's proposed contracts for the request.

[0062] In an example conveyed with respect to a particular time period t^* , requests submitted prior to time t^* may be known, contracts may have been offered and (in some cases) accepted, commitments may have been made to honor the accepted contracts, and resources may have been allocated to begin processing accepted requests. In this example, the provider may collect information from the prior requests about requests that are in the process of being served. For these requests submitted prior to t^* , the provider knows the customer's selection of contract (if any) and whenever a contract was selected, the resource cost assigned

to that request. The customer's contract selection may allow the provider to identify one value of h , and possibly multiple values of k . The types inferred from the customer's contract selection for request j may be denoted as $h(j)$ and $k(j)$.

[0063] Based on the aforementioned variables determined by cloud service manager 321, a number of rules (e.g., constraints) may be used by the cloud service provider to generate the cloud service menu. As described further, the rules may include a fixed work variables for prior time periods rule, a fixed completion time commitments for prior requests rule, a fixed revenue for prior jobs rule, a job completion rule, a one completion per job rule, an elapsed time rule, a maximum number of nodes rule, a node usage rule, an ex-post incentive compatibility rule, an ex-post individual rationality rule, a first job revenue constraint rule, a second job revenue constraint rule, a third job revenue constraint rule, and a minimal node usage rule, among others.

[0064] As noted, one constraint or rule may be a fixed work variables for prior time periods rule. The rule may be defined by the following:

$$x(j, h, k, t) = x.\text{fixed}(j, h, k, t) \text{ for all } t < t^*, j \text{ for which } s(j) < t^*.$$

For time periods prior to the current time t^* , node allocations that were made may not be changed. However, node allocations may change for periods $t \geq t^*$ involving jobs that arrived prior to t^* , as long as duration commitments for those jobs are honored. Further, information about cost type revelation $h(j)$ for jobs in the process of being served may be used to improve resource utilization.

[0065] If a contract was accepted for job j , the following is true:
 for $(h, k) = (h(j), k(j))$, $x.\text{fixed}(j, h, k, t) = x(j, h(j), k(j), t)$ determined in the solution of the model for t^*-1 .
 $x.\text{fixed}(j, h, k, t) = 0$ for all $(h, k) \neq (h(j), k(j))$
 If no contract was accepted for job j , then $x.\text{fixed}(j, h, k, t) = 0$ for all h, k, t .

[0066] Another constraint or rule may be the fixed completion time commitments for prior requests rule. The rule may be defined as follows:

$$y(j, h, k, t) = y.\text{fixed}(j, h, k, t) \text{ for all } j \text{ for which } s(j) < t^*, \text{ for all } t,$$

$u(j,h,k,d) = u.fixed(j,h,k,d)$ for all j for which $s(j) < t^*$, for all d .

For jobs submitted prior to t^* , the completion time commitments that were made cannot be changed.

- [0067]** If a contract was accepted for job j , the following is true:
for $(h,k) = (h(j),k(j))$, $y.fixed(j,h,k,t)=0$ for all completion times t except the one selected by the customer, which has $y.fixed(j,h,k,t)=1$
 $y.fixed(j,h,k,t)=0$ for all $(h,k) \neq (h(j),k(j))$
- [0068]** If no contract was accepted for job j , then $y.fixed(j,h,k,t)=0$ for all h,k,t .
- [0069]** If a contract was accepted for job j , the following is also true:
for $(h,k) = (h(j),k(j))$, $u.fixed(j,h,k,d)=0$ for all duration d except the one selected by the customer, which has $u.fixed(j,h,k,d)=1$
 $u.fixed(j,h,k,d)=0$ for all $(h,k) \neq (h(j),k(j))$.
- [0070]** If no contract was accepted for job j , then $u.fixed(j,h,k,d)=0$ for all h,k,d .
- [0071]** A fixed revenue for prior jobs rule or constraint may be defined as follows:
 $v(j,h,k,d) = v.fixed(j,h,k,d)$ for all j for which $s(j) < t^*$, for all d .
- [0072]** For requests submitted prior to t^* , the revenue to be earned from those requests cannot change.
- [0073]** If a contract was accepted for job j , the following is true:
for $(h,k) = (h(j),k(j))$, $v.fixed(j,h,k,d)=0$ for all duration d except the one selected by the customer, which has $v.fixed(j,h,k,d)=$ the revenue $v(j,h(j),k(j),d)$ determined in the solution of the model for t^*-1 .
 $v.fixed(j,h,k,d)=0$ for all $(h,k) \neq (h(j),k(j))$.
- [0074]** If no contract was accepted for job j , then $v.fixed(j,h,k,d)=0$ for all h,k,d .
- [0075]** Another constraint or rule may be the job completion rule. The job completion rule may be defined by the following:
 $\sum_{t:t(s(j)) \leq t \leq t} x(j,h,k,t) \geq r(j,h) * y(j,h,k,t)$ for each j,h,k and t where $s(j)+l(j)-1 \leq t \leq s(j) + m(j)-1$.

A job j of cost type h for customer type k may not be completed by period t unless at least $r(j, h)$ node-periods have been scheduled between $s(j)$ (the earliest start time for job j) and t .

[0076] Another constraint or rule may be the one completion per job rule. The one completion per job rule may be defined by the following:

$\sum_{t: s(j) \leq t \leq s(j) + m(j) - 1} y(j, h, k, t) \leq 1$ for each j, h, k . Job j for cost type h and customer type k may be completed in at most one time period t .

[0077] Also, the elapsed time constraint or rule may be defined by the following:

$u(j, h, k, t - s(j) + 1) = y(j, h, k, t)$ for each j, h, k and t where $s(j) + l(j) - 1 \leq t \leq s(j) + m(j) - 1$.

If job j is completed at time t for cost type h and customer type k , then its duration (relative to arrival time $s(j)$) is $t - s(j) + 1$.

[0078] The maximum number of nodes constraint or rule may be defined by the following:

$g(j, t) \geq x(j, h, k, t)$ for each j, h, k and t where $s(j) + l(j) - 1 \leq t \leq s(j) + m(j) - 1$.

The maximum number of nodes assigned to job j in period t over all cost types and customer types may be at least the number assigned to customer type k and cost type h .

[0079] The node usage constraint or rule may be defined by the following:

$\sum_{j: t \geq s(j)} g(j, t) \leq N$ for each period t .

[0080] The ex-post incentive compatibility constraint or rule may be defined by the following:

$w(j, k, d) - p(s(j), d) r(j, h) \geq w(j, k, dd) - p(s(j), dd) r(j, h) - M^*(1 - u(j, h, k, d))$ for each j, h, k, d, dd for which $l(j) \leq d \leq m(j)$ and $l(j) \leq dd \leq m(j)$ and $dd \neq d$, where M is a positive number.

If request j of cost type h is completed for customer type k in duration d , then $u(j, h, k, d) = 1$ and the last right-hand-side term $M^*(1 - u(j, h, k, d))$ equals zero. In this case the user's surplus $w(j, k, d) - p(s(j), d) r(j, h)$ may be at least his surplus $w(j, k, dd) - p(s(j), dd) r(j, h)$ under any other duration assignment dd , for $dd \neq d$. Note, if job j of

cost type h is not completed for customer type k in duration d , then $u(j,h,k,d)=0$ and the right-hand-side may be a negative number.

[0081] The ex-post individual rationality constraint or rule may be defined by the following:

$$w(j,k,d) - v(j,h,k,d) \geq 0 - M^*(1-u(j,h,k,d)) \text{ where } M \text{ is a very large number, for each } j,h,k, \text{ and } d \text{ for which } 1(j) \leq d \leq m(j).$$

If job j is completed for customer type k and cost type h in duration d , then $u(j,h,k,d)=1$ and the term $M^*(1-u(j,h,k,d))$ equals zero. In such an example, it may be that the customer's surplus $w(j,k,d) - v(j,h,k,d)$ is nonnegative.

[0082] The job revenue constraints1 rule may be defined by the following:

$$v(j,h,k,d) \leq p(s(j),d)r(j,h) \text{ for each } j,h,k, \text{ and } d \text{ for which } l(j) \leq d \leq m(j).$$

The revenue earned for job j and customer type k for completion duration d may not exceed the per-node-period price $p(s(j),d)$ charged for jobs arriving in period $s(j)$, times the number of node periods $r(j,h)$.

[0083] The job revenue constraints2 rule may be defined by the following:

$$v(j,h,k,d) \geq p(s(j),d)r(j,h) - M^*(1-u(j,h,k,d)) \text{ for each } j,h,k, \text{ and } d \text{ for which } l(j) \leq d \leq m(j).$$

If job j is completed for customer type k and cost type h in duration d , then $u(j,h,k,d)=1$ and the term $M^*(1-u(j,h,k,d))$ equals zero. In this case, the revenue earned for request j and customer type k for completion duration d is greater than or equal to the per-node-period price $p(s(j),d)$ charged for jobs arriving in period $s(j)$ and completed in d periods, times the number of node-periods $r(j,h)$. Combined with the job revenue constraints1, this constraint means that the revenue earned for request j and customer type k for completion duration d equals $p(s(j),d) r(j,h)$.

[0084] The job revenue constraints3 rule may be defined by the following:

$$v(j,h,k,d) \leq w(j,k,d) * u(j,h,k,d) \text{ for each } j,h,k \text{ and } d \text{ for which } 0 \leq d \leq m(j).$$

The revenue earned for job j of cost type h and customer type k for completion duration d may be zero if the job j of cost type h for customer k is not completed in duration d . Moreover, the revenue earned for job j of cost type h and customer

type k for completion duration d may not exceed the customer's willingness to pay for that duration.

[0085] The minimal node usage rule may be defined by the following:

$$\sum_{t: t \geq s(j)} x(j, h, k, t) \leq r(j, h) \text{ for each } j, h, k.$$

[0086] Further, variable domain rules define the domains for each of the variables described herein, and may be as follows:

$$x(j, h, k, t): \in \{0, 1, 2, \dots, N\}$$

$$y(j, h, k, t): \in \{0, 1\}$$

$$g(j, t) \text{ continuous between } 0 \text{ and } N$$

$$u(j, h, k, d) \text{ continuous between } 0 \text{ and } 1$$

$$v(j, h, k, d) \text{ continuous, nonnegative}$$

$$p(t, d) \text{ continuous, nonnegative.}$$

[0087] Interaction between each customer and the cloud resource provider in any period t^* can be described as follows. In an example, at time t^* , a number of requests j arrive. These are requests for which $s(j)=t^*$. The cloud provider sets the menu of per-node-period prices for the jobs arriving t^* , indexed by duration: $\{p(t^*, d)\}$. Given the menu $\{p(t^*, d)\}$ and the customer's willingness to pay $w(j, k, d)$, the customer submitting request j with arrival time t^* may decide whether to buy $r(j, h)$ nodes or not and, if the customer buys, the customer selects the completion duration d (by selecting one price $p(t^*, d)$ from the menu). In this example, the quantity bought by the customer is either $r(j, h)$ or 0. If the customer makes a selection, the provider learns $r(j, h)$. Further, after the selection is made by the customer, the provider may decline to serve a request if the provider so chooses.

[0088] Examples may include a customer choosing duration d by selecting $p(t^*, d)$, and in response, the provider committing to completing the request within d periods, with revenue $v(j, k, h, d)$ is realized. The value $x(j, h, k, d)$ may then be determined. Such value is the same for all customer types k that are compatible with the selection of $p(t^*, d)$. In another example, the customer chooses nothing, no revenue is realized, and $x(j, h, k, d)=0$ for all h, k, d .

[0089] One example of objective function is the maximization of the expected revenue from all requests submitted in or after t^* . Therefore, the objective function may be defined by the following:

$$\text{Maximize } z = \sum_{j,k,h,d:s(j) \geq t^*} \mu(j,h) * \pi(j,k) * v(j,k,h,d)$$

Using the variables determined, the rules provided above, and the mixed integer program solver 322, the above objective function may be determined. In other words, an allocation of resources may be determined in order to complete all cloud service requests received from users 324, while maximizing profit for the cloud service provider. For instance, revenue may be maximized from jobs arriving in periods t^* and later; the revenue from jobs arriving previously may be fixed based on prior commitments. The program solver 322, using the objective function, simultaneously makes pricing and resource allocation decisions since these are intertwined. The resource allocation decisions for jobs arriving in the future may be conditional on customer and cost type information not yet known. For a job j that arrived in t^* and prior to t^* that are not completed by t^* , the provider may know the customer type $k(j)$ and cost type $h(j)$ and can use this information to his benefit at time t^* . The provider may choose a different resource allocation for j than he had chosen previously, as long as he meets commitments for the job completion time.

[0090] Examples are not limited to the constraints listed above, and additional and/or different constraints may be employed. These extra constraints or rules may allow the cloud provider to take into account specific characteristics of the cloud system, special requests from the customers, or other factors. For example, the same mechanism can be used to derive prices that do not vary depending on the completion duration d , but vary only in terms of each request's active periods: $p(s(j))$. In that case, the ex-post incentive compatibility constraint drops (whereas the ex-post individual rationality constraint may hold), and the customer decides how many nodes to buy for each period. The protocol that regulates the interaction between the cloud provider and user with job j may be the following:

1. The provider sets $p(t^*)$.

2. The customer with job j reveals the customer's cost type h , if, given the price $p(t^*)$, the customer feels that he may be willing to buy the service.
 - a. Once h is revealed, then the total expense is known $p(t)r(j,h)$.
 - b. Learning $r(j,h)$ (and in light of the resources available at time $s(j)=t^*$), the provider decides whether to make a proposal over d . The values $x(j,h,k,t)$ for all k compatible with the payment of $p(t)r(j,h)$ and all $t \geq t^*$ are derived.
 - c. If the provider makes a proposal, given the value d proposed, the customer can compare $p(t)r(j,h)$ to the customer's willingness to pay $w(j,k,d)$ and decide whether to accept the provider's proposal.

[0091] An alternative interaction may be, in each period t^* :

1. The provider sets $p(t^*)$.
2. The customer with job j reveals his cost type h and proposes a completion time d .
 - a. Once h is revealed, then the total expense is known $p(t)r(j,h)$.
 - b. Learning $r(j,h)$ and the customer desired completion time d , the provider may decide whether to accept the proposal from the customer in light of the resources available at time $s(j)=t^*$.

[0092] In some examples, if the price is $p(t)$, the provider can reject a request after learning type h (by setting a high d in the first interaction protocol, or by just rejecting the customer's proposal in the second protocol). This may not happen when the price is $p(t,d)$. The additional price discrimination opportunities obtained by setting different prices for different request completion times may imply a lower ability to reject requests ex-post (i.e. after learning the true realization of the request cost type h).

[0093] Furthermore, in some examples, the per-resource-period price may be independent of time. To enforce this restriction, all price variables $p(t,d)$ may be replaced with a single price variable p . Put another way, the per-resource-period price is no longer dependent on time t and duration d .

[0094] Another extension may be to add a constraint on the maximum number of nodes $a(j)$ that a job can use in a given period. Such a constraint or rule may be represented by the following:

$$x(j,h,k,t) \leq a(j) \text{ for each } j,h,k \text{ and } t \text{ for which } s(j) \leq t \leq s(j) + m(j)-1.$$

[0095] In examples where requests demand that a minimum number of nodes are used for the entire duration between the time that a job starts processing until it finishes, the following variables and constraints may be employed. First, an additional variable may be included: $i(j,h,k,t)$, which is a binary variable indicating whether a positive number of nodes is assigned to job j of cost type h at time t for customer type k . This variable may be defined for t such that $s(j) \leq t \leq s(j) + m(j)-1$. Second, three additional constraints or rules may be included. The three additional constraints or rules may be defined as follows. A work indicator constraint or rule may be defined as:

$$x(j,h,k,t) \leq N * i(j,h,k,t) \text{ for each } j,h,k, \text{ and } t \text{ where } s(j) \leq t \leq s(j) + m(j)-1.$$

This constraint forces the indicator variable i to be positive if x is positive, and forces x to be zero if i is zero.

[0096] Also, a minimum number of nodes constraint may be defined as:

$$x(j,h,k,t) \geq b(j) * i(j,h,k,t) \text{ for each } j,h,k, \text{ and } t \text{ where } s(j) \leq t \leq s(j) + m(j)-1$$

and $b(j)$ is the exogenous minimum number of nodes used in each period.

This constraint forces at least $b(j)$ nodes to be used in any period where nodes are used for a job.

[0097] Additionally, a contiguous work constraint may be defined as:

$$i(j,h,k,t) \geq i(j,h,k,tt) + i(j,h,k,ttt) - 1 \text{ for each } j,h,k, \text{ and } t \text{ where } s(j) + 1 \leq t \leq s(j) + m(j)-2, \text{ and } tt < t \text{ and } ttt > t.$$

If work is done for a job j and types k and h in earlier period tt and later period ttt , then work must also be done in period t .

[0098] In some instances, it may be desired that processing of a job starts right away. This can be implemented by using the variables $i(j,h,k,t)$ in the following set of constraints:

$$u(j,h,k,d) \leq i(j,h,k,s(j)) \text{ for each } j,h,k, \text{ and duration } d \text{ where } l(j) \leq d \leq m(j).$$

This constraint indicates that job j may not be completed in duration d (for any duration d) unless the job is started in the period in which it arrives, $s(j)$.

[0099] Examples of the present disclosure include an assumption that a request j can begin processing as soon as it arrives (at time $s(j)$). This assumption, in some examples, may be relaxed for distinguishing between the time that a request becomes known $c(j)$ versus the time the request can begin processing, $s(j) \geq c(j)$.

[00100] In a different example, the variables $x(j,k,h,t)$ may be replaced with variables $x(j,t)$ which do not depend on customer type k or job type h . In this case, the $x(j,t)$ represents the maximum number of nodes allocated to job j in time t . With respect to non-reversible parameters, the following may be maintained:

$$x.\text{fixed}(j,t) \text{ for all } t < t^*, j \text{ for which } s(j) < t^*$$

If a contract was accepted for request j , then the provider is informed of the cost type h for request j , and the associated resource cost $r(j,h)$. The provider can then use this information to determine the resource assignments for job j in each period.

$$x.\text{fixed}(j,t) = x(j,t) \text{ determined in the solution of the model for } t^*-1. \text{ If } \sum_{t: s(j) \leq t < t^*} x(j,t) > r(j,h), \text{ then the provider can reduce } x.\text{fixed}(j,t) \text{ for the largest } t < t^* \text{ until } \sum_{t: s(j) \leq t < t^*} x.\text{fixed}(j,t) = r(j,h).$$

If no contract was accepted for request j , then $x.\text{fixed}(j,t)=0$ for all t .

[00101] With respect to the example in which node assignment variables are independent of job and customer type, the fixed work variables for prior time period rule may be adjusted such that the rule may be defined by the following:

$$x(j,t) = x.\text{fixed}(j,t) \text{ for all } t < t^*, j \text{ for which } s(j) < t^*.$$

The variables $g(j,t)$ may be no longer needed in an example in which node assignment variables are independent of job type and customer type. Moreover, a maximum number of nodes constraint may not be present in this. Other constraints and instances may include the removal of h and k from x variables. In some examples, a job may include a condition to use a constant number of nodes over time if it is served. For instance:

$$x(j,t) = g = x(j,tt) + x(j,ttt) - 1 \text{ for all } j \text{ and for all } t, tt \text{ and } ttt \text{ for which } tt \leq t-1 \text{ and } ttt \geq t+1.$$

[00102] Figures 4A-4D illustrate example menus for price, completion time, and resource allocation determination for cloud services according to the present disclosure. Such menus may be visible to a customer, for instance, via a user interface (e.g., graphical user interface). Figure 4A illustrates an example menu 430 including three completion times for a cloud service request, and each completion time is offered with the associated price. For instance, an 8 hour completion time is associated with a twenty cent per node hour price.

[00103] Figure 4B illustrates an example menu 432 for a cloud service request for a period different than that of Figure 4A. Figure 4B illustrates dependence on an arrival time (referred to in the menu 432 as “earliest start time”, and is shown as arrival time dependent. For instance, an arrival time of 12:00 AM with a one-hour service duration has a price per node cost of two dollars, whereas an arrival time of 1:00 AM has a price per node of twenty cents more, or two dollars and twenty cents. As such, the price per node is time-dependent.

[00104] Figure 4C illustrates an example menu 434 for a cloud service request independent of the duration of service (e.g., dynamic resource price). As shown in the example menu 434 of Figure 4C, the price per node hour is dependent only on the arrival time. Figure 4D illustrates an example menu 436 for a cloud service request independent of job arrival time and duration (e.g., fixed resource price). As shown in the example menu 436 of Figure 4D, the price per node hour depends on nothing; it is fixed.

[00105] Figure 5 illustrates an example flow chart of a method 540 for pricing of cloud resources according to the present disclosure. At 542, the method 540 can include continuously receiving, by a cloud service manager, a plurality of cloud service requests from a plurality of users in a cloud system. As discussed in relation to Figure 3, users 324 may submit cloud service requests online to the cloud service manager 321. In some examples, the method 430 can include receiving a list of cloud service requests that the cloud service provider needs to

consider. For example, each of the plurality of cloud service requests may be indexed, the indexing which may assist in the assignment of resources to complete each service request by particular completion times.

[00106] At 544, the method 540 can include determining by the cloud service manager, a plurality of per-resource-period prices for a subset of the plurality of cloud service requests based on availability of resources in the cloud system and information previously gathered with respect to the plurality of users and the plurality of cloud service requests, wherein each per-resource-period price is associated with a cloud resource allocation. The users may be past users and/or future users. For instance, for future users, the cloud provider may use beliefs over their service requests and willingness to pay. In some examples, a plurality of cloud service requests may be received, but the cloud service manager may only determine per-resource-period prices for some (e.g., a subset) of the cloud service requests received. For example, the user may request a particular cloud service, such as a Monte Carlo simulation. A plurality of different per-resource-period prices may be determined for the Monte Carlo simulation. Based on the type and amount of cloud service requests received and expected to be received in the future by the cloud service manager, various options for per-resource-period prices may be provided for each requested cloud service.

[00107] At 546, the method 548 can include offering, by the cloud service manager, to a first user among the plurality of users, a first menu of the plurality of per-resource prices for a cloud service request among the plurality of cloud service requests. For instance, a user may be presented with a menu that includes three pricing options: \$2.00 per node hour to complete the service in 1 hour, \$0.20 per node hour to complete the service in 8 hours, and \$0.05 to complete the service in 24 hours. In some instances, the cloud menu may include additional information.

[00108] Further, at 548, the method 540 can include allocating, by the cloud service manager, a first set of cloud resources to fulfill a selected service request by the first user among the plurality of users of the cloud system. For example, if a user among the plurality of users of the cloud system purchased a particular

pricing option for a requested cloud service, resources in the cloud system may be allocated to specifically perform that requested cloud service during the agreed upon time period. Notably, while the cloud resources may be reserved for a particular cloud service, there may be instances when the actual cloud service is not performed. Put another way, the user may simply reserve the cloud resources (e.g., nodes) but not actually perform an actual workload. In such instances, the user may use the cloud resources for any purpose they chose, including leaving them idle for the duration of the cloud service for the duration of the reservation time.

[00109] The method 430 may further include offering the first menu based on based on a probability distribution over requested job types, a probability distribution over possible user types of the first user, and a willingness to pay associated with possible user types of the first user. For example, once all price options are determined for each of the cloud service requests received, the cloud service manager may present a menu of pricing options to each of the users, as discussed in relation to Figure 3.

[00110] The method 540 may also include offering, by the cloud service manager, to a second user among the plurality of users, a second menu of the plurality of per-resource prices for a cloud service request among the plurality of cloud service requests and allocating, by the cloud service manager, a second set of cloud resources to fulfill a selected service request by the second user among the plurality of users of the cloud system. In such an example, the first menu and the second menu are different based on the information previously gathered and, for each of the first and the second users, a probability distribution over a requested job type, a probability distribution over a user type, and a willingness to pay associated with a user type. For instance, based on information previously gathered and due to the online nature, as new information is received, offerings may be updated.

[00111] Method 540 may be performed iteratively. For instance, a plurality of requests may be received by a cloud service manager continuously. As these

requests are received, offers associated with service duration and per-resource prices may be determined by the provider dynamically. These offers take into consideration information gathered previously, as the method 540 is performed iteratively. This information is continuously and/or dynamically updated, as the method 540 continues to be performed.

[00112] Examples of the present disclosure may allow for practical implementation of dynamic pricing and resource allocation in the sale or distribution of resource-based cloud services in the context of customers or users whose willingness-to-pay depends on their request's completion time, and this willingness to pay is the customers/users' private information. Such examples may be utilized in the sale of public cloud resources and/or the allocation of private cloud resources across different organizations in an enterprise.

[00113] As noted, examples of the present disclosure may be applied within an enterprise to distribute cloud resources (i.e. the nodes) across different departments or business units. In this case, prices $p(j,d)$ are transfer prices that are charged to the departments that rent the nodes. Instead of maximizing revenue (or profits), it may be more appropriate to select the prices in order to implement the allocation of nodes that maximizes efficiency, or, in other words, the summation of the business unit's utilities. In each period t^* , the enterprise cloud resources manager maximizes the following:

$$\text{Maximize } zz = \sum_{j,k,h,d:s(j) \geq t^*} \pi(j,k) * \sum_{j,k,h,d:s(j) > t^*} \mu(j,h) * v(j,k,h,d).$$

[00114] As compared to other approaches to pricing and resource allocation in selling IT services on-demand, examples of the present disclosure consider the following: a dependence of customers' willingness to pay on their requests completion time; the provider's uncertainty about customers' willingness to pay; the provider's uncertainty about resource demand of each customer's request; the allowance of per-resource-period price charged to a job to vary with the active time and the completion duration of a request; and an online environment in which resource requests become known over time. The desired (e.g., optimal) prices are

chosen to maximize the expected revenue of the service provider in the context of the uncertainty in willingness to pay and request resource costs.

[00115] Other approaches to resource-based sales of information technology services may not consider the customers' willingness to pay as a function of completion time, nor any uncertainty therein, and thus assume customers are indifferent between completion times for their requests. In contrast, examples of the present disclosure consider that a customer may have jobs that are urgent (deadline-sensitive batch applications), along with other jobs, such as long analytical processing jobs, that are not urgent. Examples of the present disclosure may allow a provider to better price nodes and allocate resources to maximize revenue.

[00116] In the present disclosure, reference is made to the accompanying drawings that form a part hereof, and in which is shown by way of illustration how a number of examples of the disclosure can be practiced. These examples are described in sufficient detail to enable those of ordinary skill in the art to practice the examples of this disclosure, and it is to be understood that other examples can be used and that process, electrical, and/or structural changes can be made without departing from the scope of the present disclosure.

[00117] The figures herein follow a numbering convention in which the first digit corresponds to the drawing figure number and the remaining digits identify an element or component in the drawing. Elements shown in the various figures herein can be added, exchanged, and/or eliminated so as to provide a number of additional examples of the present disclosure. In addition, the proportion and the relative scale of the elements provided in the figures are intended to illustrate the examples of the present disclosure, and should not be taken in a limiting sense. As used herein, the designators "N", and "P", particularly with respect to reference numerals in the drawings, indicate that a number of the particular feature and/or component so designated can be included with a number of examples of the present disclosure. The designators "N" and "P" can refer to a same feature and/or component, or different features and/or components.

[00118] As used herein, “logic” is an alternative or additional processing resource to perform a particular action and/or function, etc., described herein, which includes hardware, e.g., various forms of transistor logic, application specific integrated circuits (ASICs), etc., as opposed to computer executable instructions, e.g., software firmware, etc., stored in memory and executable by a processor. Further, as used herein, “a” or “a number of” something can refer to one or more such things. For example, “a number of widgets” can refer to one or more widgets. Also, as used herein, “a plurality of” something can refer to more than one of such things.

[00119] The above specification, examples and data provide a description of the method and applications, and use of the system and method of the present disclosure. Since many examples can be made without departing from the spirit and scope of the system and method of the present disclosure, this specification merely sets forth some of the many possible embodiment configurations and implementations.

What is claimed:

1. A system, comprising:
 - a request engine capable of continuously receiving requests for cloud services from a plurality of users of a cloud system;
 - an output engine to determine a plurality of variables relating to each of the continuously requested cloud services using a plurality of parameters received;
 - a menu engine to generate a cloud service menu to be offered to the users in response to a subset of the received requests to be offered to the plurality of users, the cloud service menu defining a price per-resource-period for each of the requested cloud services based on the determined variables; and
 - a resource allocation engine to allocate cloud resources to fulfill the requested cloud service in response to a user selection from the cloud service menu.
2. The system of claim 1, further comprising an optimization engine to dynamically define the price per-resource-period as new requests for cloud services are received.
3. The system of claim 1, wherein the plurality of users and the cloud service provider are entities within a same organization and the per-resource-period price is a transfer price that a user within the plurality of users is charged for the requested cloud services within the organization.
4. The system of claim 1, wherein the per-resource-period price is a price per node-period reserved for the requested cloud service.
5. The system of claim 1, further comprising the resource allocation engine to determine whether to accept the user selection based on available resources.

6. The system of claim 1, wherein the price per-resource depends on at least one of the job arrival time and the service duration.

7. A non-transitory machine-readable medium storing instructions executable by a processing resource to cause a computing system to:

continuously receive from a plurality of users of a cloud system, a plurality of cloud service requests;

receive a plurality of parameters relating to the plurality cloud service requests from at least one of the user and a provider of the cloud system;

provide a cloud menu to a subset of the plurality of users for completion of a subset of the plurality of cloud service requests, the cloud menu including a price-per resource for each of the subset of cloud service requests determined using the plurality of parameters and information gathered from previously provided price-per resource options; and

allocate cloud resources to fulfill a user-selected price-per resource option from the cloud menu.

8. The medium of claim 7, including instructions executable to receive a probability that a cloud service request among the plurality of cloud service requests is of a particular job type.

9. The medium of claim 7, including instructions executable to determine a maximum number of nodes to allocate to each cloud service request based on a determined job type.

10. The medium of claim 7, including instructions executable to receive a probability that a cloud service request among the plurality of cloud service requests is received from a particular user type.

11. The medium of claim 7, wherein the price per-resource depends on at least one of a job arrival time and a service duration.

12. A method, comprising:

continuously receiving, by a cloud service manager, a plurality of cloud service requests from a plurality of users in a cloud system;

determining by the cloud service manager, a plurality of per-resource prices for a subset of the plurality of cloud service requests based on availability of resources in the cloud system and information previously gathered with respect to the plurality of users and the plurality of cloud service requests, wherein each per-resource price is associated with a cloud resource allocation;

offering, by the cloud service manager, to a first user among the plurality of users, a first menu of the plurality of per-resource prices for a cloud service request among the plurality of cloud service requests; and

allocating, by the cloud service manager, a first set of cloud resources to fulfill a selected service request by the first user among the plurality of users of the cloud system.

13. The method of claim 12, wherein the method is performed iteratively.

14. The method of claim 12, including offering the first menu based on a probability distribution over requested job types, a probability distribution over possible user types of the first user and a willingness to pay associated with the first user.,

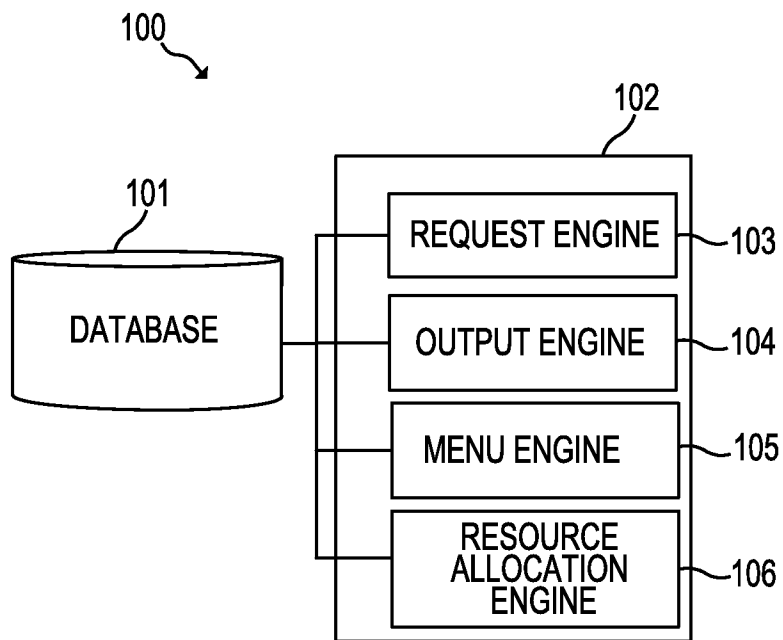
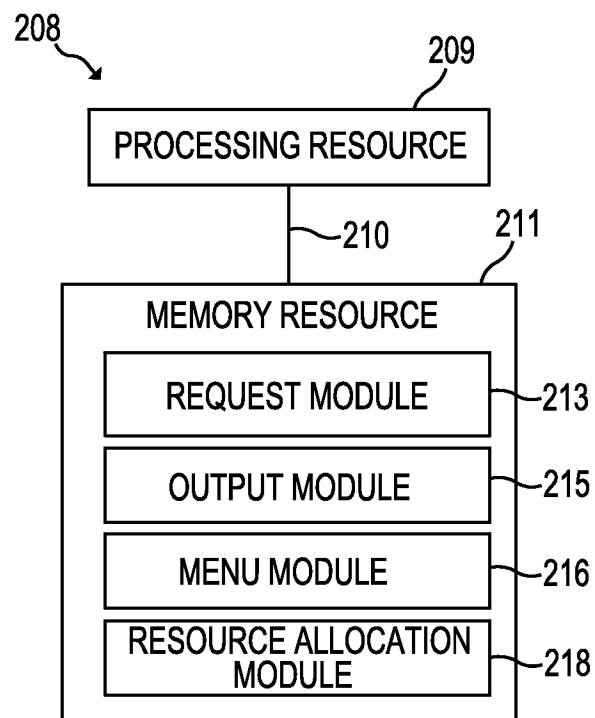
15. The method of claim 12, further comprising:

offering, by the cloud service manager, to a second user among the plurality of users, a second menu of the plurality of per-resource prices for a cloud service request among the plurality of cloud service requests; and

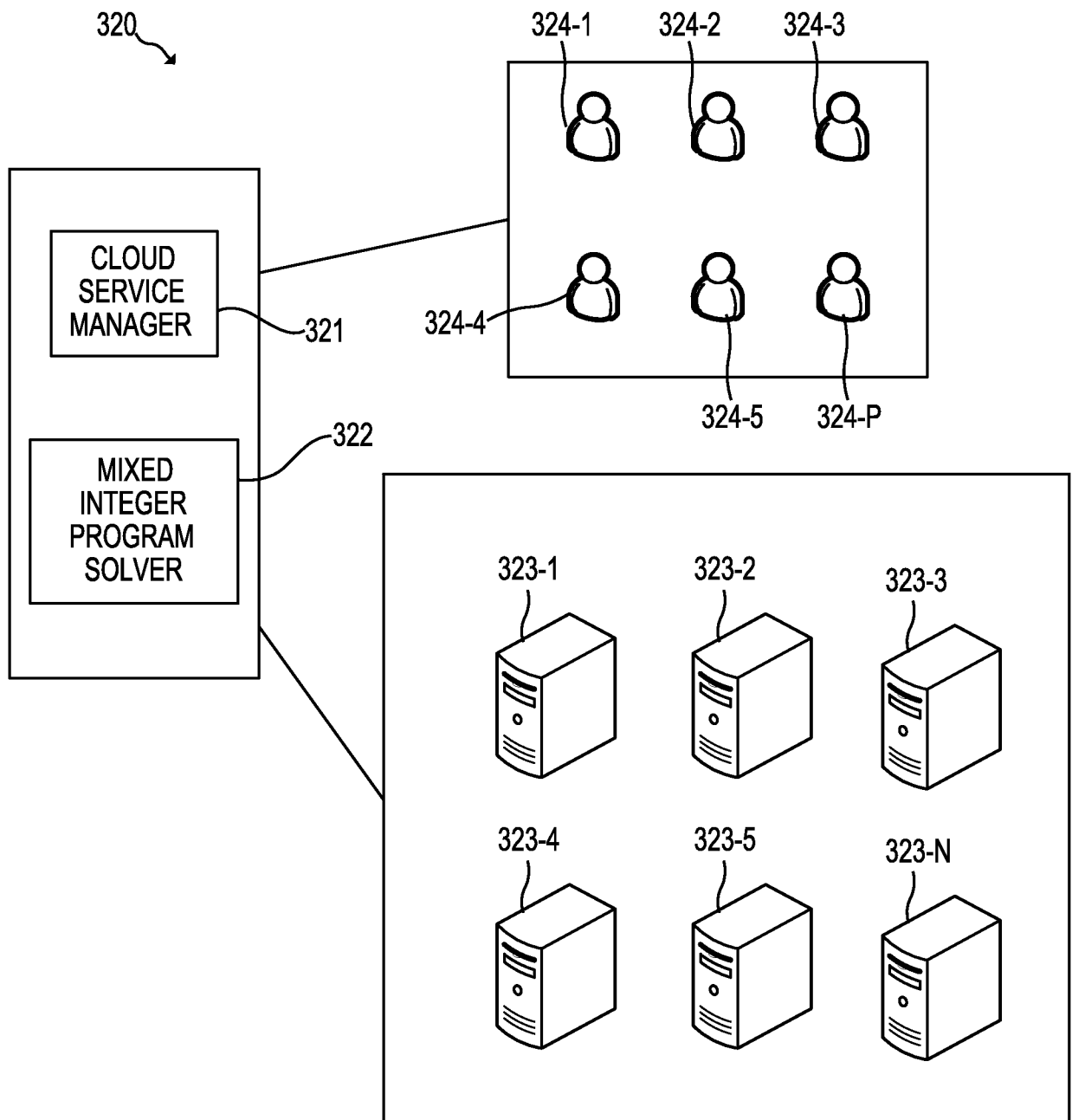
allocating, by the cloud service manager, a second set of cloud resources to fulfill a selected service request by the second user among the plurality of users of the cloud system,

wherein the first menu and the second menu are different based on the information previously gathered and, for each of the first and the second users, a probability distribution over a requested job type, a probability distribution over a user type, and a willingness to pay associated with a user type.

1/4

**Fig. 1****Fig. 2**

2/4

**Fig. 3**

3/4

430

DURATION OF SERVICE	PRICE PER NODE HOUR
1 HOUR	\$2.00
8 HOURS	\$0.20
24 HOURS	\$0.05

Fig. 4A

432

EARLIEST START TIME	DURATION OF SERVICE	PRICE PER NODE HOUR
MAY 22, 12AM	1 HOUR	\$2.00
MAY 22, 12AM	8 HOURS	\$0.20
MAY 22, 12AM	24 HOURS	\$0.05
MAY 22, 1AM	1 HOUR	\$2.20
MAY 22, 1AM	8 HOURS	\$0.22
MAY 22, 1AM	24 HOURS	\$0.06

Fig. 4B

434

EARLIEST START TIME	PRICE PER NODE HOUR
MAY 22, 12AM	\$0.50
MAY 22, 1AM	\$0.55

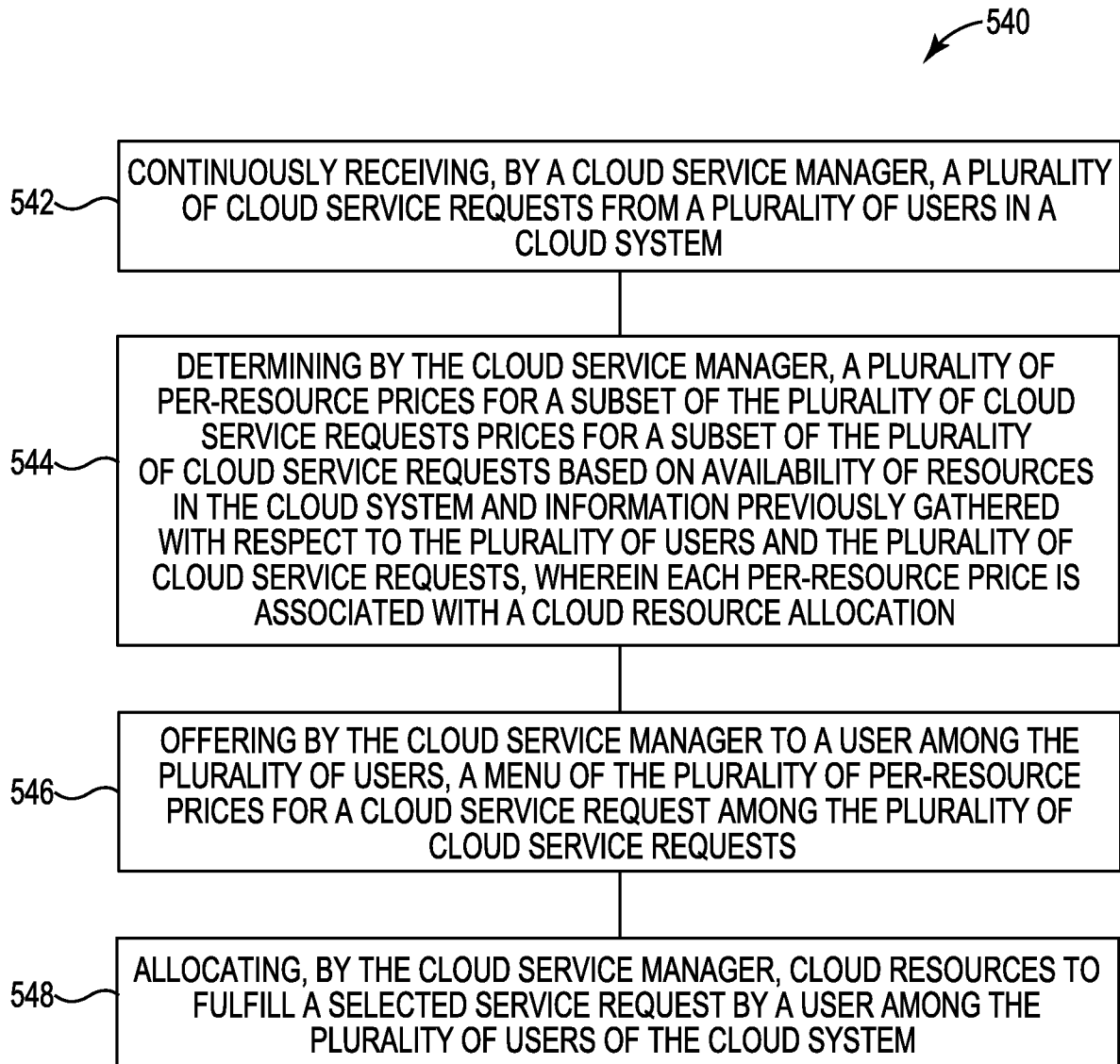
Fig. 4C

436

PRICE PER NODE HOUR
\$0.50

Fig. 4D

4/4

**Fig. 5**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/034431**A. CLASSIFICATION OF SUBJECT MATTER****G06Q 30/06(2012.01)i, G06Q 50/30(2012.01)i, G06Q 50/10(2012.01)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)

G06Q 30/06; G06F 15/173; G06F 17/00; G06F 9/44; G06Q 30/00; G06F 9/46; G06Q 10/00; G06Q 30/08; G06F 9/50; G06Q 50/30; G06Q 50/10

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, service, request, variable, parameters, price, menu, allocate

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2013-0346227 A1 (NAVENDU JAIN et al.) 26 December 2013 See paragraphs [0016],[0024]-[0025],[0034]-[0035],[0037], claims 4,17 and figures 1-3.	1-15
Y	US 2012-0095940 A1 (CHRISTIAN H. BORGS et al.) 19 April 2012 See paragraphs [0033],[0037],[0057],[0065]-[0066], claims 1,7,15-16 and figure 1.	1-15
Y	US 2012-0016721 A1 (JOSEPH WEINMAN) 19 January 2012 See abstract, claims 4-6 and figure 2B.	8
A	US 2013-0160018 A1 (DHANWANT SINGH KANG et al.) 20 June 2013 See abstract, claims 1-2,9 and figures 3-4.	1-15
A	US 2006-0059492 A1 (CRAIG W. FELLESTEIN et al.) 16 March 2006 See paragraphs [0039],[0053],[0058],[0060],[0073], claims 1,3,5,7-8,10,12 and figure 7.	1-15
A	KR 10-2014-0118030 A (INHA-INDUSTRY PARTNERSHIP INSTITUTE) 08 October 2014 See abstract, claims 1-3,6-7 and figure 1.	1-15



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

06 April 2016 (06.04.2016)

Date of mailing of the international search report

08 April 2016 (08.04.2016)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

CHO, KI YUN

Telephone No. +82-42-481-5655



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/034431

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013-0346227 A1	26/12/2013	WO 2013-192059 A2 WO 2013-192059 A3	27/12/2013 11/06/2015
US 2012-0095940 A1	19/04/2012	None	
US 2012-0016721 A1	19/01/2012	None	
US 2013-0160018 A1	20/06/2013	US 8832694 B2	09/09/2014
US 2006-0059492 A1	16/03/2006	US 7712100 B2	04/05/2010
KR 10-2014-0118030 A	08/10/2014	None	