



- (51) **International Patent Classification:**
G06Q 30/06 (2012.01) *G06Q 30/02* (2012.01)
- (21) **International Application Number:**
PCT/US2015/034370
- (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, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

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

[Continued on next page]

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

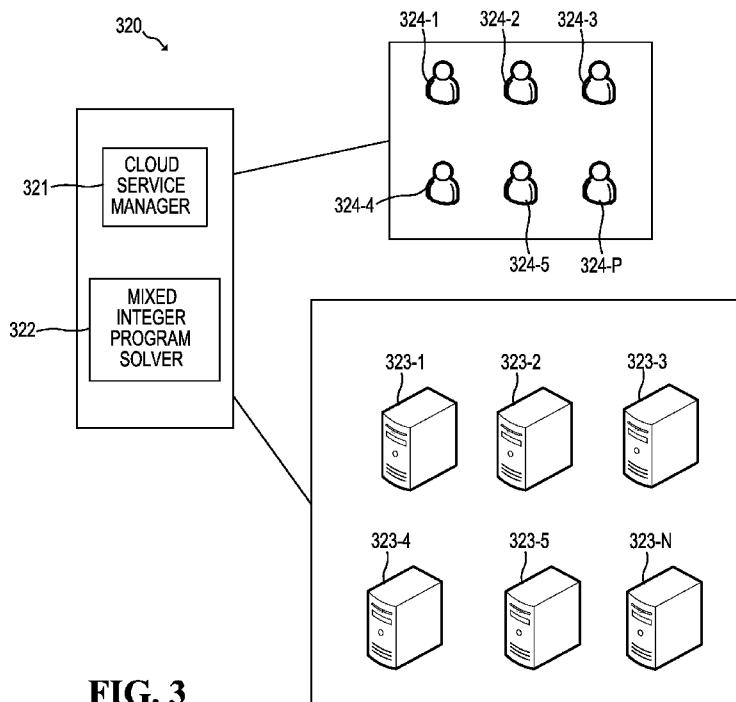


FIG. 3

(57) **Abstract:** Techniques are provided for pricing of cloud resources. A system for pricing of cloud resources can include a request engine to receive a request for a cloud service from a user of a cloud system, an output engine to determine a plurality of variables relating to the requested cloud service using a plurality of parameters received, a menu engine to generate a cloud service menu to be offered to the user, the cloud service menu defining a per-period per-resource price for the requested cloud service, and a duration of time by which the requested cloud service will be completed, 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.



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

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] Figure 4 illustrates an example flow chart of a method for pricing of cloud resources according to the present disclosure.

Detailed Description

[0006] The cloud computing model allows users (e.g., customers) to rent computing infrastructure as needed, rather than requiring them 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 requirements 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.

[0007] A batch application, herein referred to as a “service” a “job”, or a “cloud service”, may use 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.

[0008] 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 require 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.

[0009] 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).

[0010] Pricing of cloud resource in accordance with the present disclosure allows a cloud service provider to rent nodes to users for different periods of time and provide pricing options to users corresponding to the amount of time by which the nodes are rented. 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 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 requirements of the job, with typically a far shorter duration than the year or longer commitment required from reserved instances. Also, unlike the spot market model, the prices charged for the node-periods are set by the provider.

[0011] Pricing of cloud resource in accordance with the present disclosure allows for the management of the nodes (and possible readjustments over time) by the cloud provider. In accordance with the present disclosure, a user of the cloud system may buy 10 node-periods with a completion time of 3 periods and pay a price for those 10 node-periods over the 3 periods. How the node-periods are allocated to a particular job over the 3 periods (e.g. 5-5-0, 4-2-4, 4-4-2, 4-3-3,...) may be decided by the cloud provider.

[0012] Moreover, a cloud service provider may have limited resources. The service provider may face a set of potential customers requesting different services. Each service may be described in terms of an earliest starting 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). The service provider may choose how to assign resources to the service over time in order to satisfy the total resource cost requirement and the completion time selected by the customer. 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...).

[0013] Still further, pricing of cloud resources may be subject to two or more sources of uncertainty: 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 and promise different service-completion times. 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-period per-resource price, among other considerations.

[0014] 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. As used herein, a per-period per-resource 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. Examples of the

present disclosure may include systems, methods, and machine-readable and executable instructions and/or logic.

[0015] Pricing of cloud resources according to the present disclosure allows a practical implementation of the sale of resource-based 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.

[0016] 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 105, a menu engine 106, and a resource allocation engine 107). 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.).

[0017] 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).

[0018] The request engine 103 may include a combination of hardware and programming that is configured to receive a cloud service request offline from a

user (e.g., customer) of a cloud system. As used herein, a cloud service request received offline refers to a cloud service request that is submitted during a first window or time period during which users may submit cloud service requests, and which can be executed during a second window or time period. For example, a first window or time period (e.g., the “offline” window or time period) may be a period of time during which users may submit cloud service requests. Similarly, a second window or time period, such as an “execution” window or time period, may be a period of time during which the requested cloud services are executed. At the end of each “execution” window or time period, another window or time period may begin in which users may submit cloud service requests, and another “execution” window or time period may begin. As such, the cloud service request collection and execution windows may repeat a number of times. In some examples, the request-collection and execution time windows may be defined in the same way for all request services. For instance, the request-collection window may extend from Monday 00:00am -7:00am and the execution window may extend from Monday 9:00am-5:00pm.

[0019] In the foregoing examples, “time” is considered a sequence of discrete intervals (e.g. minutes, hours, days,...) during the second window, i.e., the execution time period. Given that, a set of time periods $T = \{1, \dots, T\}$ may be considered. At the beginning of the first period, the full list of cloud service requests (e.g., jobs) that the provider needs to consider may be known. The last period T may define the horizon length over which the price and resource allocation system is applied. The extension of such horizon 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.

[0020] In some examples, the system 100 may include a compilation engine (not illustrated in Figure 1). The compilation engine may include a combination of hardware and programming that is configured to compile a list of cloud service requests received from a plurality of users of the cloud system. For example, a plurality of cloud service requests may be received from a plurality of users. The

compilation engine and/or a cloud service manager (discussed further in relation to Figure 3) may determine a full list of the cloud service requests received. For example, a first user of the cloud system may request a Monte Carlo simulation to be run, and a second user of the cloud system request an optimization problem to be solved. The compilation engine may determine that a request for a Monte Carlo simulation and a request for an optimization problem have been received.

[0021] In some examples, the system 100 may include an input engine (not illustrated in Figure 1). The input engine 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.

[0022] 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 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 $r(j, h)$. Put another way, $r(j, h)$ may represent the maximum number of node periods that may be allocated to a particular cloud service request (e.g., job j) based on the determined job type. Yet another example of a parameter may be (an estimate 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) and for a given service duration d , 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 during 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 . Notably, parameters j and $s(j)$ may be entered by the customer whereas all other parameters may be entered by the

cloud provider. The parameters from the customer may be entered at the request collection time (the first time-window). The parameters from the cloud provider may be entered even before the customer enters his parameters. In other words, the system 100 may be pre-set with cloud provider parameters.

[0023] The output engine 105 may include a combination of hardware and programming that is configured to determine a plurality of variables relating to the requested cloud service using the plurality of parameters and a mixed integer programming solver. 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 values. 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 0 and 5 (e.g., 0, 1, 2.5, 4.1, etc.).

[0024] 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 nodes, a maximum number of per-period servers, 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 nodes 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 per-period per-resource price, may be determined. As used herein the per-period per-resource price 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 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.

[0025] 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-period per-resource 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-period per-resource price may be a “cost” or transfer price that the marketing department is “debited” or “charged” by the company information technology (IT) department for utilizing the cloud services and/or reserving the cloud resources associated with the requested cloud services.

[0026] The menu engine 106 may include a combination of hardware and programming that is configured to generate a cloud service menu to be offered to the user, the cloud service menu defining a per-period per-resource price for the requested cloud service based on the determined variables. Based on the values of the variables determined, various menu options may be provided to the user. For instance, the menu engine 106 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 for which the resources (e.g., nodes) are allocated to the cloud service requested, a second price (e.g., intermediate) that has a second duration of time for which the resources (e.g., nodes) are allocated to the cloud service requested, and a third price (e.g., the highest) that has a third duration of time for which the resources (e.g., nodes) are allocated to the cloud service requested. 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.

[0027] 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, and 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 and/or reserve resources for the agreed upon period of time and for the agreed upon price reflected in the cloud menu.

[0028] The resource allocation engine 107 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 107 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 107 may reject the user selection from the cloud service menu. Further, the resource allocation engine 107 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.

[0029] 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.

[0030] 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.

[0031] 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.

[0032] A number of modules (e.g., request module 213, input module 214, 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 input 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.).

[0033] 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. In another example, the input module 214 may include instructions that when executed by the processing resource 209 may function as the input engine 104. Additionally, the output module 215 may include instructions that when executed by the processing resource 209 may function as the output engine 105.

[0034] The request module 213 may include instructions that when executed by the processing resource 209, may receive from a plurality of users of a cloud system, a plurality of cloud service requests. As described in relation to Figure 1, a single cloud service request may be received from a single user of the cloud system, or a plurality of cloud service requests may be received (e.g., from a single user and/or from a plurality of users of the cloud system).

[0035] The input module 214 may include instructions that when executed by the processing resource 209, may receive parameters relating to the requested cloud service from at least one of the user and a provider of the cloud system. 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.

[0036] The computing device 208 may include a menu module 216 that when executed by the processing resource 209 may provide a cloud menu to the

plurality of users for completion of the plurality of cloud service requests, the cloud menu including a per-period per-resource price for each of the cloud service requests.

[0037] The menu module 216 may include instructions that when executed by the processing resource 209 may provide a cloud menu to the plurality of users for completion of the plurality of cloud service requests, the cloud menu including a per-period per-resource price for each of the cloud service requests, the per-period per-resource prices associated with the duration of time and determined using the plurality of parameters. In some examples, the menu module 216 may provide a cloud menu to the plurality of users for completion 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.

[0038] 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 per-period per-resource price option from the cloud menu within the duration of time.

[0039] In some examples, the computing device 208 may further include instructions that when executed by the processing resource 209 may determine a job type for each cloud service request received from users of the cloud system, and determine 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 require 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 determine 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 per-period per-resource price 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.

[0040] 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.

[0041] The cloud service manager 321 may deploy and manage cloud services. For instance, the cloud service manager 321 may receive requests offline for cloud services from a plurality of users 324-1, ...324-P (e.g., customers) of a cloud system. Put another way, at time 0 all requests for cloud services j may be collected. Each service request may also be referred to as a "job". Before time period 1, the full set of jobs may be known. 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. Put another way, cloud service manager 321 may determine a plurality of resources 323 for each cloud service request.

[0042] As discussed previously herein, each job $j \in J$ may have the following characteristics:

$s(j)$: the earliest period in which the job j can start. This might correspond to the job's arrival time or to the time the job j may begin.

$r(j, h)$: 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 .

[0043] Each service request (e.g., job) may be interpreted as a separate customer with a given willingness to pay 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 during d periods after arrival time $s(j)$.

[0044] 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 .

[0045] 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 $T = \max_i \{s(j) + m(j)\}$ to ensure enough time in the horizon to complete every job received from the users 324.

[0046] The cloud service manager 321 may receive a plurality of cloud service requests from users 324. The cloud service manager 321 may determine a

full list of cloud service requests from the plurality of users 324 of the cloud system. Based on the full list of cloud service requests received, the cloud service manager 321 may determine a plurality of resources 323 for each cloud service request received.

[0047] Based on the cloud service requests received from the users 324, the cloud service manager 321 may determine a plurality of variables 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 servers 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, is $p(s(j), d) r(j, h)$.

[0048] 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 required 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.

[0049] Based on the determined variables, 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 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.

[0050] One constraint or rule may be the job completion rule. The job completion rule may be defined by the following:

$$\sum_{t: s(j) \leq t \leq t} x(j, h, k, t) \geq r(j, h) * y(j, h, k, t) \text{ for each } j, h, k \text{ and } t \text{ where } s(j) + l(j) - 1 \leq t \leq s(j) + m(j) - 1.$$

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 .

[0051] Another constraint or rule may be the one completion per job rule. The one completion per job rule may be defined by the following:

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

[0052] Also, the elapsed time constraint or rule may be defined by the following:

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

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 $t-s(j)+1$.

[0053] The maximum number of nodes constraint or rule may be defined by the following:

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

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 .

[0054] The node usage constraint or rule may be defined by the following:

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

[0055] The ex-post incentive compatibility constraint or rule may be defined by the following:

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

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.

[0056] The ex-post individual rationality constraint or rule may be defined by the following:

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

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 such an example, it may be required that the customer's surplus $w(j, k, d) - v(j, h, k, d)$ is nonnegative.

[0057] 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 required number of node periods $r(j, h)$.

[0058] 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 required 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 $p(s(j), d) r(j, h)$.

[0059] 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.

[0060] The minimal node usage rule may be defined by the following:

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

[0061] Further, variable domain rules define the domains for each of the variables described herein, and may be as follows:

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

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

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

$u(j, h, k, d)$ continuous between 0 and 1

$v(j, h, k, d)$ continuous, nonnegative

$p(t, d)$ continuous, nonnegative.

[0062] One example of an objective function is to maximize the expected revenue from all jobs. Therefore, the objective function may be defined by the following:

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

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.

[0063] In some examples, the variables $x(j, h, k, t)$ may be replaced with variables $x(j, t)$ where $x(j, t)$ represents the maximum number of nodes allocated to job j in time t . In such examples, the $x(j, t)$ variable no longer depends on cost type h and customer type k . In such instances, the $g(j, t)$ variables are no longer needed in the model, and are replaced with $x(j, t)$ in the node usage constraints.

Furthermore, the maximum number of node constraint are not needed, and $x(j, h, k, tt)$ may be replaced with $x(j, tt)$ in the job completion constraint.

[0064] For further clarification, the protocol for pricing of cloud resources according to the present disclosure may be the following:

1. Sequence of requests is given (exogenously).
2. The cloud provider sets the menu of per-period per-resource prices: $\{p(s(j), d)\}$.
3. Given the menu $\{p(s(j), d)\}$ and the customer's willingness to pay $w(j, k, d)$ the customer submitting request j with active time $s(j)$ decides whether to buy $r(j, h)$ nodes or not and, if he buys, he selects the completion duration d (e.g., by selecting one price $p(s(j), d)$ from the menu). The quantity of node-periods bought by the customer is either $r(j, h)$ or 0.

4. Given the available resources over time, and the duration d selected by the customer, the cloud resource provider decides whether to accept the customer's purchase of $r(j, h)$. Otherwise, revenue $v(j, k, h, d)$ is realized.

[0065] 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 incentive compatibility constraint is non-existent. The protocol that regulates the interaction between the cloud provider and user in such examples may be the following.

1. Sequence of requests is given (exogenously)
2. The cloud provider sets $p(t)$
3. Each user reveals h .
 - 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 expected resource available at time $s(j)$) the cloud provider decides whether to make an offer over completion duration d for the requested cloud service.
 - c. If the cloud provider makes an offer, given the value d offered, the user can compare $p(t)r(j, h)$ to his wtp $w(j, k, d)$ and decide whether to accept the cloud provider's offer.

[0066] Furthermore, in some examples, the per-period per-resource price may be fixed over 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-period per-resource price is no longer dependent on time t or duration d . Notably, the cloud provider still retains the freedom to allocate nodes to services by scaling up and/or down resource allocations (e.g., increasing and/or decreasing resource allocations) so long as after d periods the service is completed. Put another way, the cloud

provider may increase or decrease the allocated cloud resources for completion of the requested cloud service within a specified duration of time (e.g., d periods).

[0067] 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:

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

[0068] In examples where requests require 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:

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

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

[0069] Also, a minimum number of nodes constraint may be defined as:

$$x(j,h,k,t) \geq b(j) \cdot i(j,h,k,t) \text{ for each } j,h,k, \text{ and } t \text{ where } s(j) \leq t \leq s(j) + m(j)-1 \text{ and } b(j) \text{ is the exogenous minimum number of nodes used in each period.}$$

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

[0070] 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 .

[0071] Moreover, pricing of cloud resources in accordance with the present disclosure can be applied within an enterprise to distribute cloud resources (i.e. the

nodes) across different departments or business units. In such examples, prices $p(t,d)$ may refer to transfer prices that are charged to the departments that rent the nodes. Instead of maximizing 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 units' utilities. The objective function in this example may be represented by the following:

$$\text{Maximize } zz = \sum_{j,k,h,d} \mu(j, h) * \pi(j, k) * w(j, k, d) * u(j, h, k, d)$$

[0072] Figure 4 illustrates an example flow chart of a method 430 for pricing of cloud resources according to the present disclosure. At 431, the method 430 can include 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 offline 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.

[0073] At 432, the method 430 can include determining a plurality of per-period per-resource prices for the plurality of cloud service requests based on availability of resources in the cloud system, wherein each per-period per-resource price is associated with a cloud resource allocation. In some examples, a plurality of cloud service requests may be received, but the cloud service manager may only determine per-period per-resource 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-period per-resource prices may be determined for the Monte Carlo simulation. Based on the type and amount of cloud service requests received by the cloud service manager, various options for per-period per-resource prices may be provided for each requested cloud service.

[0074] At 433, the method 430 can include offering by the cloud service manager to a user among the plurality of users, a menu of the plurality of per-period 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, such as the earliest start time of a service. Also, as described herein, the cloud menu may include only 1 (one) per-period per-resource price and associated duration for the requested cloud service.

[0075] Further, at 434, the method 430 can include allocating, by the cloud service manager, cloud resources to fulfill a selected service request by a user 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.

[0076] The method 430 may further include providing a menu of service price options to the customer based on availability of resources, customer willingness to pay for each service, and the cloud service requests received. 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.

[0077] 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.

[0078] 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.

[0079] 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.

[0080] 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 to receive a request for a cloud service from a user of a cloud system;
 - an output engine to determine a plurality of variables relating to the requested cloud service using a plurality of parameters received;
 - a menu engine to generate a cloud service menu to be offered to the user, the cloud service menu defining a per-period per-resource price for the requested cloud service, and a duration of time by which the requested cloud service will be completed, 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, wherein the per-period per-resource price is defined based on the duration of time and an earliest start time of the requested cloud service.
3. The system of claim 1, wherein the user and the cloud service provider are entities within a same organization and the per-period per-resource price is a transfer price that the user is charged for the requested cloud services within the organization.
4. The system of claim 1, further comprising the resource allocation engine to revise the allocation of the cloud resources in response to input from the cloud provider, to increase or decrease the allocated cloud resources for completion of the requested cloud service within the duration of time.

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, further comprising a selection engine to record the user selection of a particular menu item in the cloud service menu.
7. The system of claim 1, further comprising the resource allocation engine to determine that the user has declined the offer in the cloud service menu after a threshold period of time has elapsed and a user selection has not been received.
8. A non-transitory machine-readable medium storing instructions executable by a processing resource to cause a computing system to:
 - receive from a plurality of users of a cloud system, a plurality of cloud service requests;
 - receive a plurality of parameters relating to the cloud service requests from at least one of the user and a provider of the cloud system;
 - provide a cloud menu to the plurality of users for completion of the plurality of cloud service requests, the cloud menu including a per-period per-resource price for each of the cloud service requests, and a duration of time by which each of the cloud service requests will be completed, the per-period per-resource price associated with the duration of time and determined using the plurality of parameters; and
 - allocate cloud resources to fulfill a user-selected per-period per-resource price option from the cloud menu within the duration of time.
9. The medium of claim 8, wherein the instructions to receive a plurality of parameters relating to the cloud service requests from at least one of the user and a provider of the cloud system include instructions to receive from the provider, a probability that a cloud service request among the plurality of cloud service requests is of a particular job type.

10. The medium of claim 9, wherein the instructions to receive a plurality of parameters relating to the cloud service requests from at least one of the user and a provider of the cloud system include instruction to receive from the provider, a maximum number of per-period nodes to allocate to each cloud service request based on the determined job type.

11. The medium of claim 8, wherein the instructions to receive a plurality of parameters relating to the cloud service requests from at least one of the user and a provider of the cloud system include instructions to receive from the provider, a probability that a cloud service request among the plurality of cloud service requests is received from a particular user type.

12. The medium of claim 8, including instructions 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.

13. A method, comprising:

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-period per-resource prices for the plurality of cloud service requests based on availability of resources in the cloud system, wherein each per-period per-resource price is associated with a duration of time by which the requested cloud service will be completed;

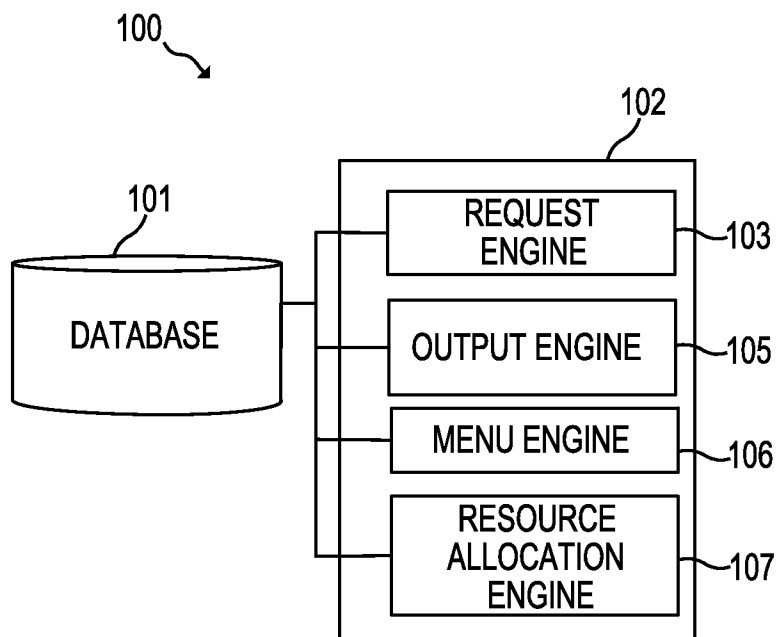
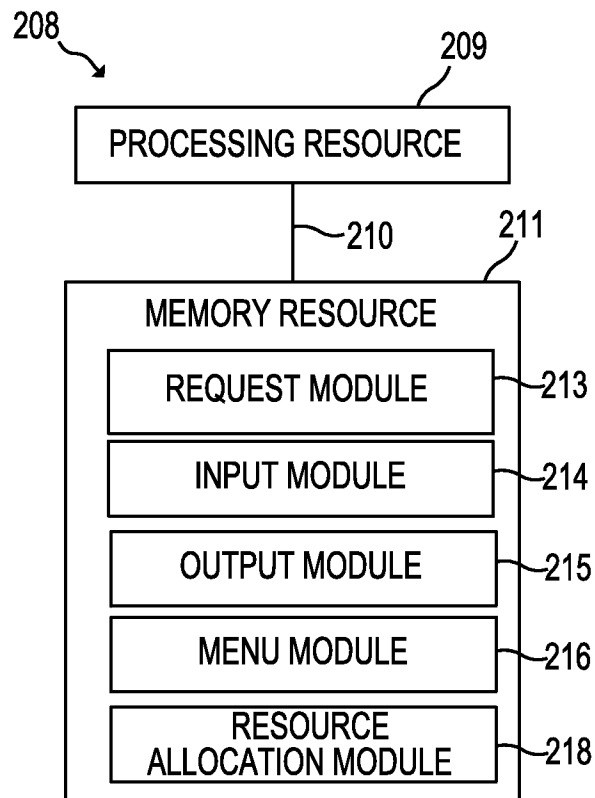
offering by the cloud service manager to a user among the plurality of users, a menu of the plurality of per-period per-resource prices for a cloud service request among the plurality of cloud service requests; and

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

14. The method of claim 13, including receiving a list of cloud service requests for the cloud service provider to consider.

15. The method of claim 13, wherein the menu of the plurality of per-period per-resource prices is based on availability of cloud resources, customer willingness to pay, and the cloud service requests received.

1/3

**FIG. 1****FIG. 2**

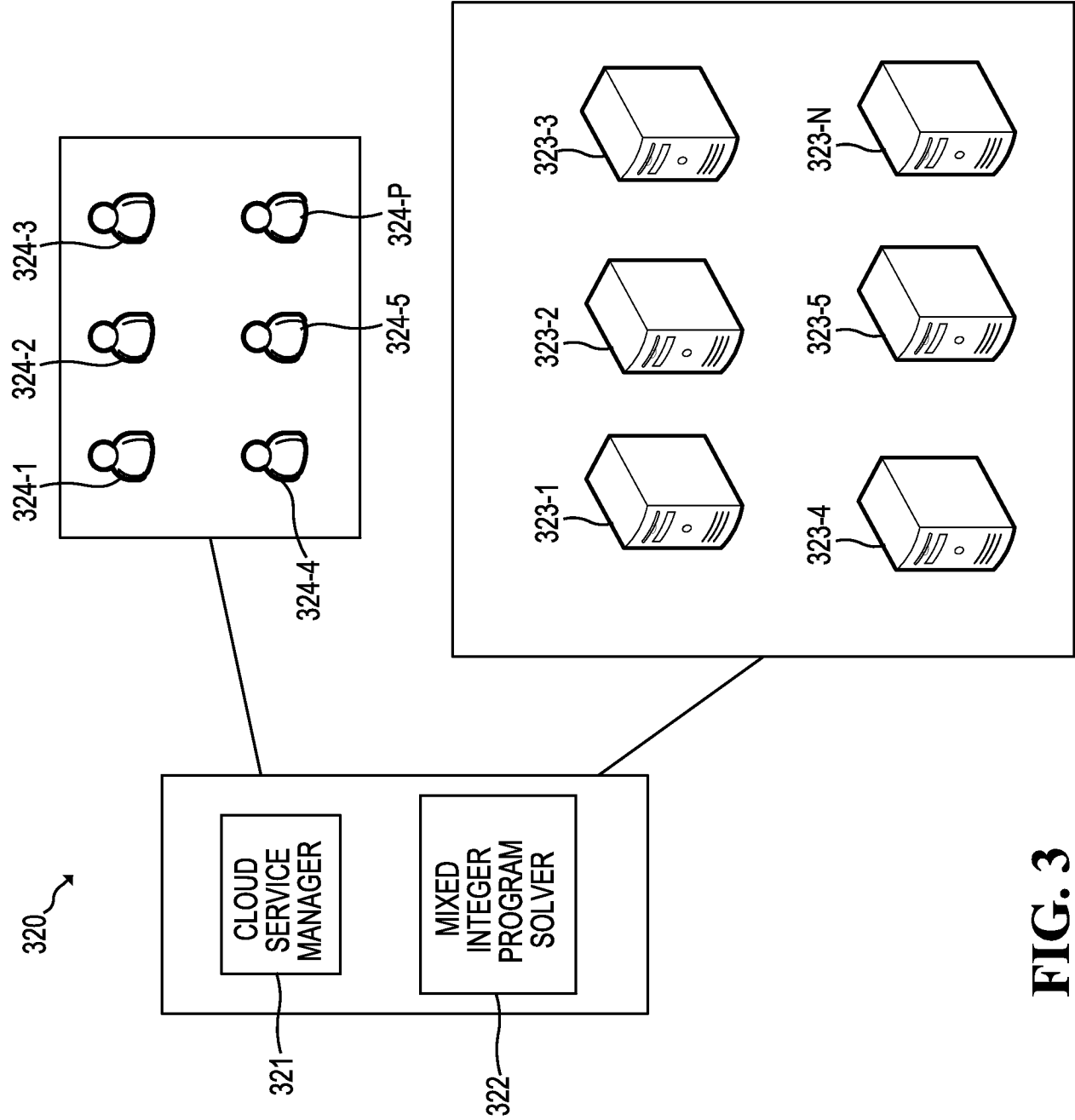
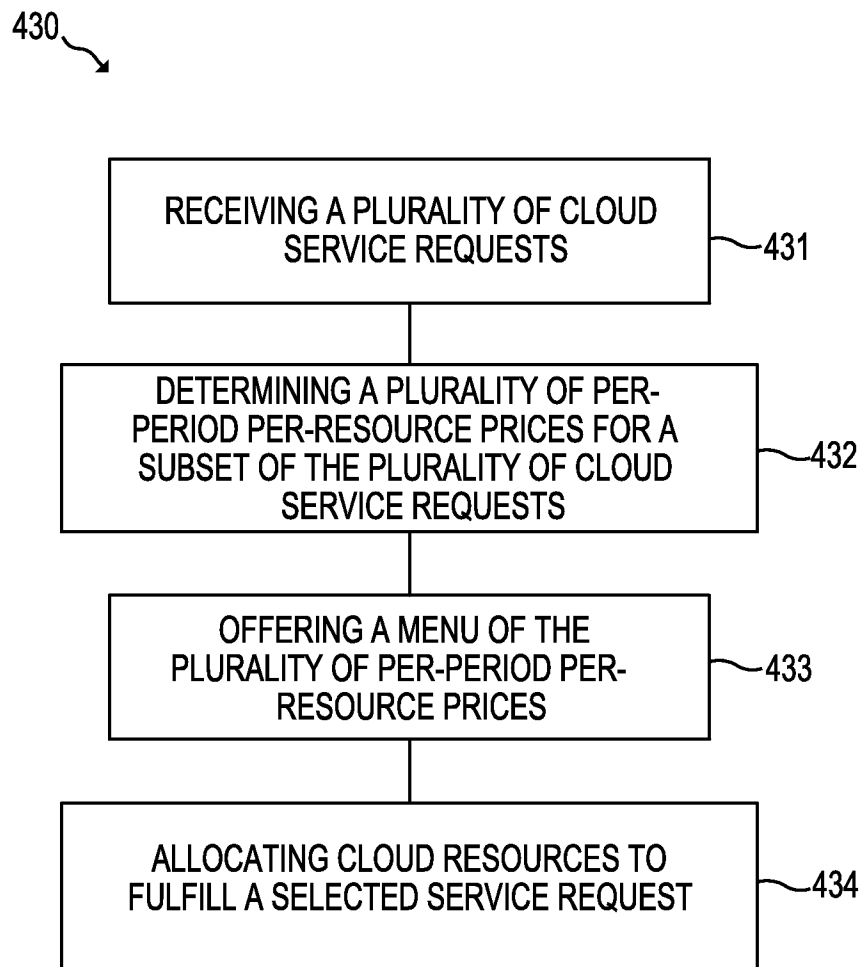


FIG. 3

3/3

**FIG. 4**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/034370**A. CLASSIFICATION OF SUBJECT MATTER****G06Q 30/06(2012.01)i, G06Q 30/02(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 30/02

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, variables, parameters, menu, per-period, per-resource, price, availability, duration, 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 abstract, paragraphs [0016],[0019],[0025],[0034]-[0035],[0037],[0043]-[0044], claims 2,4,17 and figures 1,3.	1-15
Y	US 2012-0095940 A1 (CHRISTIAN H. BORGS et al.) 19 April 2012 See abstract, paragraphs [0033],[0037],[0057],[0066], claims 1,7,15-16 and figure 1.	1-15
Y	US 2012-0016721 A1 (JOSEPH WEINMAN) 19 January 2012 See paragraph [0041], claims 4-6 and figure 2B.	9-10,14
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/US2015/034370

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 2006-0059492 A1	16/03/2006	US 7712100 B2	04/05/2010
KR 10-2014-0118030 A	08/10/2014	None	