



(51) International Patent Classification:  
*G06Q 10/06* (2012.01) *G06Q 30/06* (2012.01)

(21) International Application Number:  
PCT/US2015/048697

(22) International Filing Date:  
4 September 2015 (04.09.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
62/187,110 30 June 2015 (30.06.2015) US  
14/846,349 4 September 2015 (04.09.2015) US

(71) Applicant: APPTIO, INC. [US/US]; 11100 NE 8th Street,  
Suite 600, Bellevue, WA 98004 (US).

(72) Inventors: AURADKAR, Rahul V.; 11100 NE 8th Street,  
Suite 600, Bellevue, WA 98004 (US). HAYMAN, Ed-  
ward, Mark; 11100 NE 8th Street, Suite 600, Bellevue,  
WA 98004 (US). TIJERINA, David; 11100 NE 8th  
Street, Suite 600, Bellevue, WA 98004 (US). KRISH-

NAN, Venkatesh; 11100 NE 8th Street, Suite 600, Bel-  
levue, WA 98004 (US).

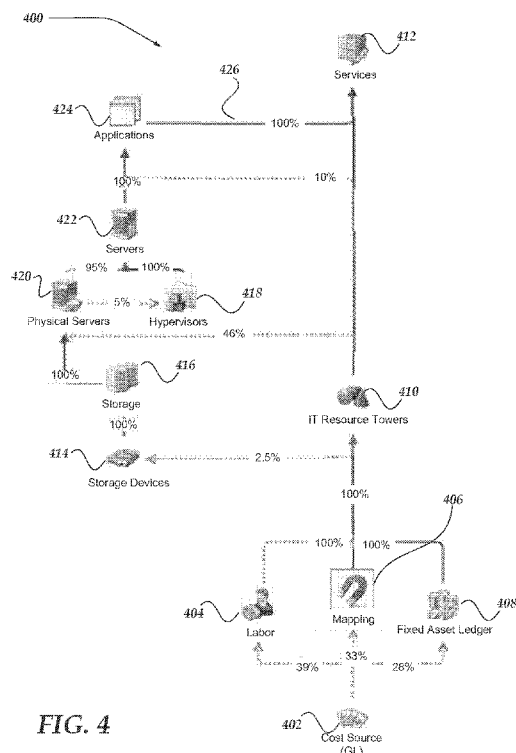
(74) Agent: BRANCH, John W.; Loowe Graham Jones PLLC,  
701 Fifth Avenue, Suite 4800, Seattle, WA 98104 (US).

(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,

[Continued on next page]

(54) Title: INFRASTRUCTURE BENCHMARKING BASED ON DYNAMIC COST MODELING



(57) Abstract: Embodiments are directed towards infrastructure benchmarking based on dynamic cost modeling. Raw customer models that represent a customer's business system may be ingested into a benchmarking system and mapped to customer models based on one or more standard models. Benchmarking information that may be generated from one or more community models that are one or more customer models of other customers that may be available to the benchmarking application. Additional third-party information, such as, industry survey information may also be used for generating benchmarking information. The benchmarking information may be used to generate reports that include comparisons between one or more customer models and the one or more community models. The benchmarking information may be continuously updated based on changes to customer models and/or data, community models and/or data, external information.



---

LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, **Published:**  
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, — *with international search report (Art. 21(3))*  
GW, KM, ML, MR, NE, SN, TD, TG).

**INFRASTRUCTURE BENCHMARKING BASED ON DYNAMIC COST  
MODELING**

**CROSS-REFERENCE TO RELATED APPLICATIONS**

5           This application is a Utility Patent application based on a previously filed U.S. Provisional Patent Application U.S. Serial No. 62/187,110 filed on June 30, 2015, entitled “INFRASTRUCTURE BENCHMARKING BASED ON DYNAMIC COST MODELING,” the benefit of the filing date of which is hereby claimed under 35 U.S.C. §119(e) and which is further incorporated by reference in its entirety.

**TECHNICAL FIELD**

10           The present invention relates generally to computer automated activity based budget modeling, forecasting and cost accounting, and more particularly, but not exclusively to generating benchmarks for evaluating budget modeling, forecasting and cost accounting.

**BACKGROUND**

15           Businesses that strive to remain viable and successful in today’s competitive commercial environment are required to adopt accurate and responsive budgeting practices. To improve efficiency, businesses use financial models that apply modern budgeting, forecasting and cost accounting techniques. For some accounting techniques, the complexity of the financial allocation model may increase as the number of tracked activities and  
20           elements increases. Therefore, for larger enterprises, sophisticated computer programs and computing devices are often required to assist in generating useful and relevant budgets based on financial allocation models. In some cases the complexity of the models and the modelled items and entities may make it difficult to compare efficiencies across large enterprises and/or between different enterprises. Thus, it is with respect to these considerations and  
25           others that the invention has been made.

**BRIEF DESCRIPTION OF THE DRAWINGS**

30           Non-limiting and non-exhaustive embodiments of the present invention are described with reference to the following drawings. In the drawings, like reference numerals refer to like parts throughout the various figures unless otherwise specified. For a better understanding of the present invention, reference will be made to the following Description of the Various Embodiments, which is to be read in association with the accompanying drawings, wherein:

FIGURE 1 illustrates a system diagram showing components of an environment in which at least one of the various embodiments may be practiced;

FIGURE 2 shows one embodiment of a client computer that may be included in a system;

5        FIGURE 3 illustrates one embodiment of a network computer that may be included in a system;

FIGURE 4 shows one embodiment of a model for automatically generated business system;

FIGURE 5 illustrates a table that may include dataset information;

10       FIGURE 6 shows a logical architecture of a model for an automatically generated business system;

FIGURE 7 shows a portion of a data model in accordance with at least one of the various embodiments;

15       FIGURE 8 shows a portion of a logical architecture for a benchmarking system that performs actions in accordance with at least one of the various embodiments;

FIGURE 9 illustrates a logical representation of a benchmark comparison table that is in accordance with at least one of the various embodiments;

FIGURE 10 illustrates a flowchart for a process for infrastructure benchmarking based on dynamic cost modeling in accordance with at least one of the various embodiments;

20       FIGURE 11 illustrates an overview flowchart for a process that generates a customer model for infrastructure benchmarking in accordance with at least one of the various embodiments;

FIGURE 12 illustrates a flowchart for a process for mapping raw customer model elements to a customer model in accordance with at least one of the various embodiments;

25       FIGURE 13 illustrates a flowchart for a process for generating benchmarking information in accordance with at least one of the various embodiments;

FIGURE 14 illustrates a logical schematic of a system for generating benchmark value for a customer in accordance with at least one of the various embodiments;

FIGURE 15 illustrates graphical examples of benchmark components that may be generated by a benchmark component generator in accordance with at least one of the various embodiments;

5 FIGURE 16 illustrates a logical representation of a benchmark engine for server computers in accordance with at least one of the various embodiments;

FIGURE 17 illustrates a flowchart for a process for generating benchmarking values using benchmarking engines in accordance with at least one of the various embodiments;

FIGURE 18 illustrates a flowchart for a process for generating benchmarking values using benchmarking engines in accordance with at least one of the various embodiments;

10 FIGURE 19 illustrates a user interface for displaying benchmarking information in accordance with at least one of the various embodiments;

FIGURE 20 illustrates a user interface for displaying benchmarking information in accordance with at least one of the various embodiments;

15 FIGURE 21 illustrates a user interface for displaying benchmarking information in accordance with at least one of the various embodiments;

FIGURE 22 illustrates a user interface for viewing/generating mapping rules for mapping raw customer models to customer models in accordance with at least one of the various embodiments; and

20 FIGURE 23 illustrates a user interface for reviewing mapped models in accordance with at least one of the various embodiments.

### **DESCRIPTION OF THE VARIOUS EMBODIMENTS**

The present innovations now will be described more fully hereinafter with reference to the accompanying drawings, which form a part hereof, and which show, by way of  
25 illustration, specific embodiments by which the invention may be practiced. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art. Among other things, the present invention may be embodied as  
30 methods or devices. Accordingly, the present invention may take the form of an entirely

hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects. The following detailed description is, therefore, not to be taken in a limiting sense.

Throughout the specification and claims, the following terms take the meanings explicitly associated herein, unless the context clearly dictates otherwise. The phrase “In one of the embodiments” or “in at least one of the various embodiments” as used herein does not necessarily refer to the same embodiment, though it may. Furthermore, the phrase “in another embodiment” as used herein does not necessarily refer to a different embodiment, although it may. Thus, as described below, various embodiments of the invention may be readily combined, without departing from the scope or spirit of the invention.

In addition, as used herein, the term “or” is an inclusive “or” operator, and is equivalent to the term “and/or,” unless the context clearly dictates otherwise. The term “based on” is not exclusive and allows for being based on additional factors not described, unless the context clearly dictates otherwise. In addition, throughout the specification, the meaning of “a,” “an,” and “the” include plural references. The meaning of “in” includes “in” and “on.”

As used herein, the terms “Financial allocation model,” “data model”, and “cost model” refers to a graph based representation of a system of financial allocation rules that can be used for costing actual expenditures (for management accounting) or budgeting future expenditures. Nodes in the model may represent classes of items that may be associated with costs and/or expenses. The edges of the graph may represent how the costs and/or expenses may be allocated between the nodes. A financial allocation model may be a visual rendering of a graph showing the nodes and the edges connecting the nodes.

As used herein, the term “Cost line item,” refers to a single line item in a budget (or finance allocation model) and its associated cost/expense. For example, the costs associated with a particular computer that is an email server may be a single item having a particular cost (e.g., the email server may correspond to a cost line item).

As used herein, the term “category,” refers to a set and/or class of cost line items that may be grouped together. Also, dataset information in fields of a dataset template may be mapped to one or more categories in a category template. For example, a collection of computers performing services such as email, web serving, enterprise resource planning, may represent separate cost line items and they may be grouped into the Servers category.

As used herein, the terms “allocation rules,” “entity propagation rules,” or “propagation rules” refer to rules in the financial data model that determine how the costs/expenses from a category are apportioned between/among other categories. Also, such rules may be assigned to individual cost line items. For example, if an email server cost line item has a value of \$1000 an allocation or entity propagation rule may be defined such that 50% of the expense may be allocated to the Marketing department and 50% may be allocated to the Engineering department. Also, allocation rules may be applied at the category as well as the cost line item level.

As used herein, the term “assignment ratios,” refers to an allocation rule, or the results of applying one or more rules, of the distribution ratio of costs to cost line items or categories. For example, if \$1000 may be allocated to Servers category, and the cost line item Email Server is allocated \$800 and the cost line item FTP Server is allocation \$200, the assignment ratios may be determined to 80% to budget item line Email Server and 20% to cost line item FTP Server. Assignment ratios may be explicitly defined by allocation rules. Or they may be derived from the allocation tables by converting the values into ratios of the total allocation to the category.

As used herein, the terms “business system” and/or “generated business system,” refers to a system that has been generated using the budget and forecasting platform. Various embodiments disclosed herein may be related to financial applications. But, one of ordinary skill in the art will appreciate that generated business systems are not limited to financial applications.

As used herein, the term “external data source” refers to any separately located system that may enable and/or provide access to one or more datasets of information.

As used herein, the term “dataset” refers to a collection of data, usually presented in tabular form. Each column may represent a particular variable. Each row may represent a given member of the dataset. Also, it may list values for fields for each of the variables, such as name, location, cost, owner, manufacturer, serial number, or the like. Non-tabular datasets can also take the form of marked up strings of characters, such as an XML file.

As used herein, the term “total cost value” refers to a value that is determined for providing at least one offering. A model for determining the total cost value of at least one offering is based at least on an allocation of cost data to at least one category in a category template in a cost model.

As used herein, the term “source object” refers to an object in a financial data model that may be providing values (e.g., costs/expenses) that may be allocated to one or more other objects (target objects). In at least one of the various embodiments, source objects may be used to represent one or more categories in a data model.

5           As used herein, the term “target object” refers to an object in a financial data model that may be allocated values (e.g., costs/expenses) from one or more other objects (source objects). In at least one of the various embodiments, target objects may be used to represent one or more categories in a data model.

10           As used herein, the term “raw customer model” refers to a data structure and/or dataset that describes a customer’s business systems. For example, a financial model of a company’s cost structures.

15           As used herein, the term “raw customer data” refers to the dataset associated with a raw customer model. In some cases, the raw customer data may be inherently included in the raw customer model. In some embodiments raw customer models and their corresponding raw customer data may be provided in various formats. In simple cases, raw customer models may be spreadsheets, database table information, csv files, or the like. In other cases, raw customer models may be structured XML files, models from one or more cost modeling systems, or the like.

20           As used herein, the term “standard model” refers to a model that may be used as a template and/or example to generate customer models (described below). The standard models may include taxonomies that provide a common definition/structure that facilitate generating benchmarking information. Standard models may be considered templates for financial allocation models, cost models, data models, or the like.

25           As used herein, the term “customer model” refers to a model of a customer business system that is generated by mapping a raw customer model to some or all of a standard model. The customer model is a model produced from a raw customer model that may be used in the benchmarking process. Customer models may be financial allocation models, cost models, data models, or the like.

30           As used herein, the term “community model” refers to other customer models that a benchmarking application may refer to if generating benchmarking information. Community

models are customer models for other customers. In some embodiments, a customer model may be compared to one or more community models.

As used herein, the term “model elements” refers to items that comprises a given model. The model elements comprising customer models may be considered categories in financial allocation models, cost models, or the like. Also, in some embodiments, model elements may be considered towers or sub-towers of cost models. One or more model elements in a raw customer model may be mapped to one or more elements of a customer model based on a standard model.

As used herein, the term “benchmarking information” refers to values that are computers and/or generated from various sources, such as, community models, third-party/external information, industry surveys, or the like. Benchmarking information may be ratios, ranges, averages, means, median, min/max values, time-series, regressions, or the like, or combination thereof, related to values associated with one or more model elements. A benchmarking application may be arranged to generate various types of benchmarking information depending on the benchmark types, model types, model compositions, configurations, user input, or the like, or combination thereof.

As used herein the term “benchmark engine” refers to a system that generates one or more benchmark values based on one or more inputs. Benchmark engines may be generated from benchmark information and benchmark data derived from one or more community models. Once a benchmark engine is generated benchmark values may be computed absent benchmark information and/or benchmark data.

As used herein the term “benchmark component” refers to a cost driver for a model item that may comprise the benchmark engine. For example, a benchmark component for benchmarking the total unit cost of servers may be the number of servers under management. Other benchmark components for servers may include, the number of CPUs, geographic location, average age of servers, type/brand of servers, or the like. Each benchmark component contributes to the benchmark engine for a given model item. Not every benchmark component is significant. Accordingly, during the generating of a benchmark engine one or more benchmark components may be excluded.

As used herein the term “customer benchmark value” refers to a benchmark value that is generated from a benchmark engine.

The following briefly describes the embodiments of the invention in order to provide a basic understanding of some aspects of the invention. This brief description is not intended as an extensive overview. It is not intended to identify key or critical elements, or to delineate or otherwise narrow the scope. Its purpose is merely to present some concepts in a simplified form as a prelude to the more detailed description that is presented later.

Briefly stated, various embodiments are directed towards infrastructure benchmarking based on dynamic cost modeling. In at least one of the various embodiments, raw customer models that may represent a customer's business system may be ingested into a benchmarking system and mapped to a customer model based on one or more standard models. In at least one of the various embodiments, a benchmarking application may include a mapping engine the uses one or more rules for determining how to map elements of raw customer models into customer models.

In at least one of the various embodiments, benchmarking information that may be generated from one or more community models. In at least one of the various embodiments, community models may be one or more customer models of other customers that may be available in a cloud computing environment, or the like. In at least one of the various embodiments, additional third-party information, such as, industry survey information may also be used for generating benchmarking information used in comparison and/or benchmarking comparison reports.

Further, in at least one of the various embodiments, real-time information such geolocation information, power/energy information, weather information, production information, or the like, or combination thereof may be collected and used in part to generate the benchmarking information.

In at least one of the various embodiments, benchmarking information may be used to generate reports that include comparisons between one or more customer models and the one or more community models. In at least one of the various embodiments, the benchmarking information may be continuously updated based on changes to customer models and/or data, community models and/or data, external information, or the like, or combination thereof. Accordingly, in at least one of the various embodiments, the benchmarking information may provide real-time and/or near real-time comparison information for evaluating the performance of one or more customer models.

Further, in at least one of the various embodiments, resource allocations may be managed using benchmark engines that are separate from the community models. Accordingly, in at least one of the various embodiments, one or more community models maybe generated based on raw model data such that the raw model data is mapped to the one  
5 or more community models based on standard models. In at least one of the various embodiments, model items that may be represented with benchmark engines may be determined based on the one or more community models.

10 In at least one of the various embodiments, one or more benchmark components for each of the plurality of model items may be generated based on benchmark data included in the one or more community models. Also, in at least one of the various embodiments, the benchmark data included in the one or more community models may be modified based on real-time information obtained from one or more of geolocation sensors, weather information sensors, or electrical power sensors.

In at least one of the various embodiments, the one or more benchmark components may be ranked for each of the plurality of model items based on an evaluation of a their contribution to a total cost value of their corresponding model item. In at least one of the various embodiments, ranking the one or more benchmark components may further include, employing best fit analysis, such as, correlation coefficients, goodness of fit, coefficient of determination, chi-squared test, or the like, to evaluate the benchmark components for ranking. Also, in at least one of the various embodiments, generating the benchmarking data may include: determining additional relevant information from one or more external data sources; and modifying the benchmarking data based on the additional relevant information.

Accordingly, in at least one of the various embodiments, one or more benchmark engines for each of the plurality of model items may be generated based on a combination of benchmark components such that the benchmark components may be selected based on their ranking.

15 In at least one of the various embodiments, if an update to a raw model data value exceeds a threshold, benchmark engines may be re-generated based on the updated raw model data. Also, in at least one of the various embodiments, if a number of community models are added or removed from exceeds a threshold value, the benchmarking engines may be updated based on the added or removed community models.

Also, in at least one of the various embodiments, if an actual data value may be to a benchmark engine of a model item, the benchmark engine may be used to generate a benchmark values for benchmarking the model item.

## 5 **Illustrative Operating Environment**

FIGURE 1 shows components of one embodiment of an environment in which at least one of the various embodiments may be practiced. Not all of the components may be required to practice various embodiments, and variations in the arrangement and type of the components may be made. As shown, system 100 of FIGURE 1 includes local area networks ("LANs") / wide area networks ("WANs") - (network) 111, wireless network 110, client  
10 computer 101-104, and Cost Modeling Platform Server (CMP) 107.

Generally, client computers 102-104 may include virtually any portable computing device capable of receiving and sending a message over a network, such as network 111, wireless network 110, or the like. Client computers 102-104 may also be described generally  
15 as client computers that are configured to be portable. Thus, client computers 102-104 may include virtually any portable computing device capable of connecting to another computing device and receiving information. Such devices include portable devices such as, cellular telephones, smart phones, display pagers, radio frequency (RF) devices, infrared (IR) devices, Personal Digital Assistants (PDA's), handheld computers, laptop computers, wearable  
20 computers, tablet computers, integrated devices combining one or more of the preceding devices, or the like. As such, client computers 102-104 typically range widely in terms of capabilities and features. For example, a cell phone may have a numeric keypad and a few lines of monochrome Liquid Crystal Display (LCD) on which only text may be displayed. In another example, a web-enabled mobile device may have a touch sensitive screen, a stylus,  
25 and several lines of color LCD in which both text and graphics may be displayed.

Client computer 101 may include virtually any computing device capable of communicating over a network to send and receive information, including messaging, performing various online actions, or the like. The set of such devices may include devices that typically connect using a wired or wireless communications medium such as personal  
30 computers, tablet computers, multiprocessor systems, microprocessor-based or programmable consumer electronics, network Personal Computers (PCs), or the like. In at least one of the various embodiments, at least some of client computers 102-104 may operate over wired

and/or wireless network. Today, many of these devices include a capability to access and/or otherwise communicate over a network such as network 111 and/or wireless network 110. Moreover, client computers 102-104 may access various computing applications, including a browser, or other web-based application.

5 In at least one of the various embodiments, one or more of client computers 101-104 may be configured to operate within a business or other entity to perform a variety of services for the business or other entity. For example, client computers 101-104 may be configured to operate as a web server, an accounting server, a production server, an email server, video game server, an inventory server, or the like. However, client computers 101-104 are not  
10 constrained to these services and may also be employed, for example, as an end-user computing node, in other embodiments. Further, it should be recognized that more or less client computers may be included within a system such as described herein, and embodiments are therefore not constrained by the number or type of client computers employed.

A web-enabled client computer may include a browser application that is configured  
15 to receive and to send web pages, web-based messages, or the like. The browser application may be configured to receive and display graphics, text, multimedia, or the like, employing virtually any web-based language, including a wireless application protocol messages (WAP), or the like. In at least one of the various embodiments, the browser application is enabled to employ Handheld Device Markup Language (HDML), Wireless Markup  
20 Language (WML), WMLScript, JavaScript, Standard Generalized Markup Language (SGML), HyperText Markup Language (HTML), eXtensible Markup Language (XML), HTML5, or the like, to display and send a message. In at least one of the various embodiments, a user of the client computer may employ the browser application to perform various actions over a network.

25 Client computers 101-104 also may include at least one other client application that is configured to receive and/or send data, including budgeting and forecasting information, between another computing device. Client applications may include a capability to provide requests and/or receive data relating to the cost models, budget reports, budget project information, allocation rules, or the like. The client application may provide data representing  
30 assignment and/or allocation changes, selecting templates, editing cost allocations between or among categories, generating and/or modifying recursive allocation rules, or the like. In at least one of the various embodiments, client applications may receive and/or generate data

related to budgeting and financial models and may generate tables and relationships between and among the data. In at least one of the various embodiments, client computers 101-104 may view and/or modify generated data models.

Wireless network 110 is configured to couple client computers 102-104 and its components with network 111. Wireless network 110 may include any of a variety of wireless sub-networks that may further overlay stand-alone ad-hoc networks, or the like, to provide an infrastructure-oriented connection for client computers 102-104. Such sub-networks may include mesh networks, Wireless LAN (WLAN) networks, cellular networks, or the like.

Wireless network 110 may further include an autonomous system of terminals, gateways, routers, or the like connected by wireless radio links, or the like. These connectors may be configured to move freely and randomly and organize themselves arbitrarily, such that the topology of wireless network 110 may change rapidly.

Wireless network 110 may further employ a plurality of access technologies including 2nd (2G), 3rd (3G), 4th (4G), 5th (5G) generation radio access for cellular systems, WLAN, Wireless Router (WR) mesh, or the like. Access technologies such as 2G, 3G, 4G, 5G, and future access networks may enable wide area coverage for mobile devices, such as client computers 102-104 with various degrees of mobility. For example, wireless network 110 may enable a radio connection through a radio network access such as Global System for Mobil communication (GSM), General Packet Radio Services (GPRS), Enhanced Data GSM Environment (EDGE), Wideband Code Division Multiple Access (WCDMA), High Speed Downlink Packet Access (HSDPA), Long Term Evolution (LTE), or the like. In essence, wireless network 110 may include virtually any wireless communication mechanism by which information may travel between client computers 102-104 and another computing device, network, or the like.

Network 111 is configured to couple network computers with other computing devices, including, CMP 107, client computer(s) 101, and through wireless network 110 to client computers 102-104. Network 111 is enabled to employ any form of computer readable media for communicating information from one electronic device to another. Also, network 111 can include the Internet in addition to local area networks (LANs), wide area networks (WANs), direct connections, such as through a universal serial bus (USB) port, other forms of computer-readable media, or any combination thereof. On an interconnected

set of LANs, including those based on differing architectures and protocols, a router acts as a link between LANs, enabling messages to be sent from one to another. In addition, communication links within LANs typically include twisted wire pair or coaxial cable, while communication links between networks may utilize analog telephone lines, full or fractional dedicated digital lines including T1, T2, T3, and T4, Integrated Services Digital Networks (ISDNs), Digital Subscriber Lines (DSLs), wireless links including satellite links, or other communications links known to those skilled in the art. For example, various Internet Protocols (IP), Open Systems Interconnection (OSI) architectures, and/or other communication protocols, architectures, models, and/or standards, may also be employed within network 111 and wireless network 110. Furthermore, remote computers and other related electronic devices could be remotely connected to either LANs or WANs via a modem and temporary telephone link. In essence, network 111 includes any communication method by which information may travel between computing devices.

Additionally, communication media typically embodies computer-readable instructions, data structures, program modules, or other transport mechanism and includes any information delivery media. By way of example, communication media includes wired media such as twisted pair, coaxial cable, fiber optics, wave guides, and other wired media and wireless media such as acoustic, RF, infrared, and other wireless media. Such communication media is distinct from, however, processor-readable storage devices described in more detail below.

CMP 107 may include virtually any network computer usable to perform data processing operation that may be used for generating cost models, data models, allocation rules, recursive allocation rules, cost allocations, total cost values for offerings, displays and/or reports thereof, such as network computer 300 of FIGURE 3. In at least one of the various embodiments, CMP 107 employs various techniques to create, define, generate, and/or automated data processing applications such as budgeting and financial management applications and one or more cost models and/or data models. CMP 107 may include modules for generating data processing applications that may apply models that may include dataset templates, category templates, allocation rules, recursive allocation rules or the like. Furthermore, CMP 107 may include and/or generate data processing applications for visualizing the generated allocation categories, cost allocations, budgets, cost models, data models, allocation rules, recursive allocation rules, total cost values for offerings, or the like.

Devices that may operate as CMP 107 include various network computers, including, but not limited to personal computers, desktop computers, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, server devices, tablet computers, network appliances, or the like. It should be noted that while CMP 107 is illustrated as a single network computer, the invention is not so limited. Thus, in another embodiment, CMP 107 may represent a plurality of network computers. For example, in at least one of the various embodiments, CMP 107 may be distributed over a plurality of network computers and/or implemented using cloud architecture.

Moreover, CMP 107 is not limited to a particular configuration. Rather, CMP 107 may operate using a controller/worker approach over a plurality of network computers, within a cluster, a peer-to-peer architecture, cloud-based architecture (e.g., virtual machines), and/or any of a variety of other architectures. Thus, CMP Server Computer 107 is not to be construed as being limited to a single environment, and other configurations, and architectures are also envisaged. CMP Server Computer 107 may employ processes and such as described below in conjunction with FIGURE 4 and above to perform at least some of its actions.

### **Illustrative Client computer**

FIGURE 2 shows one embodiment of client computer 200 that may include many more or less components than those shown. Client computer 200 may represent, for example, at least one embodiment of mobile computers or client computers shown in FIGURE 1.

Client computer 200 may include processor 202 in communication with memory 204 via bus 228. Client computer 200 may also include power supply 230, network interface 232, audio interface 256, display 250, keypad 252, illuminator 254, video interface 242, input/output interface 238, haptic interface 264, global positioning systems (GPS) receiver 258, open air gesture interface 260, temperature interface 262, camera(s) 240, projector 246, pointing device interface 266, processor-readable stationary storage device 234, and processor-readable removable storage device 236. Client computer 200 may optionally communicate with a base station (not shown), or directly with another computer. And in one embodiment, although not shown, a gyroscope may be employed within client computer 200 to measuring and/or maintaining an orientation of client computer 200.

Power supply 230 may provide power to client computer 200. A rechargeable or non-rechargeable battery may be used to provide power. The power may also be provided by an

external power source, such as an AC adapter or a powered docking cradle that supplements and/or recharges the battery.

Network interface 232 includes circuitry for coupling client computer 200 to one or more networks, and is constructed for use with one or more communication protocols and technologies including, but not limited to, protocols and technologies that implement any portion of the OSI model for mobile communication (GSM), CDMA, time division multiple access (TDMA), UDP, TCP/IP, SMS, MMS, GPRS, WAP, UWB, WiMax, SIP/RTP, GPRS, EDGE, WCDMA, LTE, UMTS, OFDM, CDMA2000, EV-DO, HSDPA, or any of a variety of other wireless communication protocols. Network interface 232 is sometimes known as a transceiver, transceiving device, or network interface card (NIC).

Audio interface 256 may be arranged to produce and receive audio signals such as the sound of a human voice. For example, audio interface 256 may be coupled to a speaker and microphone (not shown) to enable telecommunication with others and/or generate an audio acknowledgement for some action. A microphone in audio interface 256 can also be used for input to or control of client computer 200, e.g., using voice recognition, detecting touch based on sound, and the like.

Display 250 may be a liquid crystal display (LCD), gas plasma, electronic ink, light emitting diode (LED), Organic LED (OLED) or any other type of light reflective or light transmissive display that can be used with a computer. Display 250 may also include a touch interface 244 arranged to receive input from an object such as a stylus or a digit from a human hand, and may use resistive, capacitive, surface acoustic wave (SAW), infrared, radar, or other technologies to sense touch and/or gestures.

Projector 246 may be a remote handheld projector or an integrated projector that is capable of projecting an image on a remote wall or any other reflective object such as a remote screen.

Video interface 242 may be arranged to capture video images, such as a still photo, a video segment, an infrared video, or the like. For example, video interface 242 may be coupled to a digital video camera, a web-camera, or the like. Video interface 242 may comprise a lens, an image sensor, and other electronics. Image sensors may include a complementary metal-oxide-semiconductor (CMOS) integrated circuit, charge-coupled device (CCD), or any other integrated circuit for sensing light.

Keypad 252 may comprise any input device arranged to receive input from a user. For example, keypad 252 may include a push button numeric dial, or a keyboard. Keypad 252 may also include command buttons that are associated with selecting and sending images.

Illuminator 254 may provide a status indication and/or provide light. Illuminator 254 may remain active for specific periods of time or in response to events. For example, when illuminator 254 is active, it may backlight the buttons on keypad 252 and stay on while the client computer is powered. Also, illuminator 254 may backlight these buttons in various patterns when particular actions are performed, such as dialing another client computer.

Illuminator 254 may also cause light sources positioned within a transparent or translucent case of the client computer to illuminate in response to actions.

Further, client computer 200 may also comprise hardware security module (HSM) 268 for providing additional tamper resistant safeguards for generating, storing and/or using security/cryptographic information such as, keys, digital certificates, passwords, passphrases, two-factor authentication information, or the like. In some embodiments, hardware security module may be employed to support one or more standard public key infrastructures (PKI), and may be employed to generate, manage, and/or store keys pairs, or the like. In some embodiments, HSM 268 may be arranged as a hardware card that may be added to a client computer.

Client computer 200 may also comprise input/output interface 238 for communicating with external peripheral devices or other computers such as other client computers and network computers. The peripheral devices may include an audio headset, display screen glasses, remote speaker system, remote speaker and microphone system, and the like. Input/output interface 238 can utilize one or more technologies, such as Universal Serial Bus (USB), Infrared, WiFi, WiMax, Bluetooth™, and the like.

Haptic interface 264 may be arranged to provide tactile feedback to a user of the client computer. For example, the haptic interface 264 may be employed to vibrate client computer 200 in a particular way when another user of a computer is calling. Open air gesture interface 260 may sense physical gestures of a user of client computer 200, for example, by using single or stereo video cameras, radar, a gyroscopic sensor inside a computer held or worn by the user, or the like. Camera 240 may be used to track physical eye movements of a user of client computer 200.

In at least one of the various embodiments, client computer 200 may also include sensors 262 for determining geolocation information (e.g., GPS), monitoring electrical power

conditions (e.g., voltage sensors, current sensors, frequency sensors, and so on), monitoring weather (e.g., thermostats, barometers, anemometers, humidity detectors, precipitation scales, or the like), light monitoring, audio monitoring, motion sensors, or the like. Sensors 262 may be one or more hardware sensors that collect and/or measure data that is external to client

5 computer 200

GPS transceiver 258 can determine the physical coordinates of client computer 200 on the surface of the Earth, which typically outputs a location as latitude and longitude values. GPS transceiver 258 can also employ other geo-positioning mechanisms, including, but not limited to, triangulation, assisted GPS (AGPS), Enhanced Observed Time Difference (E-  
10 OTD), Cell Identifier (CI), Service Area Identifier (SAI), Enhanced Timing Advance (ETA), Base Station Subsystem (BSS), or the like, to further determine the physical location of client computer 200 on the surface of the Earth. It is understood that under different conditions, GPS transceiver 258 can determine a physical location for client computer 200. In at least one embodiment, however, client computer 200 may, through other components, provide other  
15 information that may be employed to determine a physical location of the client computer, including for example, a Media Access Control (MAC) address, IP address, and the like.

Human interface components can be peripheral devices that are physically separate from client computer 200, allowing for remote input and/or output to client computer 200. For example, information routed as described here through human interface components such  
20 as display 250 or keyboard 252 can instead be routed through network interface 232 to appropriate human interface components located remotely. Examples of human interface peripheral components that may be remote include, but are not limited to, audio devices, pointing devices, keypads, displays, cameras, projectors, and the like. These peripheral components may communicate over a Pico Network such as Bluetooth™, Zigbee™ and the  
25 like. One non-limiting example of a client computer with such peripheral human interface components is a wearable computer, which might include a remote pico projector along with one or more cameras that remotely communicate with a separately located client computer to sense a user's gestures toward portions of an image projected by the pico projector onto a reflected surface such as a wall or the user's hand.

30 A client computer may include web browser application 226 that may be configured to receive and to send web pages, web-based messages, graphics, text, multimedia, and the like. The client computer's browser application may employ virtually any programming language, including a wireless application protocol messages (WAP), and the like. In at least

one embodiment, the browser application is enabled to employ Handheld Device Markup Language (HDML), Wireless Markup Language (WML), WMLScript, JavaScript, Standard Generalized Markup Language (SGML), HyperText Markup Language (HTML), eXtensible Markup Language (XML), HTML5, and the like.

5           Memory 204 may include RAM, ROM, and/or other types of memory. Memory 204 illustrates an example of computer-readable storage media (devices) for storage of information such as computer-readable instructions, data structures, program modules or other data. Memory 204 may store BIOS 208 for controlling low-level operation of client computer 200. The memory may also store operating system 206 for controlling the operation  
10 of client computer 200. It will be appreciated that this component may include a general-purpose operating system such as a version of UNIX, or LINUX™, or a specialized client computer communication operating system such as Windows Phone™, or the Symbian® operating system. The operating system may include, or interface with a Java virtual machine module that enables control of hardware components and/or operating system operations via  
15 Java application programs.

          Memory 204 may further include one or more data storage 210, which can be utilized by client computer 200 to store, among other things, applications 220 and/or other data. For example, data storage 210 may also be employed to store information that describes various capabilities of client computer 200. The information may then be provided to another device  
20 or computer based on any of a variety of events, including being sent as part of a header during a communication, sent upon request, or the like. Data storage 210 may also be employed to store social networking information including address books, buddy lists, aliases, user profile information, or the like. Data storage 210 may further include program code, data, algorithms, and the like, for use by a processor, such as processor 202 to execute  
25 and perform actions. In one embodiment, at least some of data storage 210 might also be stored on another component of client computer 200, including, but not limited to, non-transitory processor-readable removable storage device 236, processor-readable stationary storage device 234, or even external to the client computer.

          Applications 220 may include computer executable instructions which, when  
30 executed by client computer 200, transmit, receive, and/or otherwise process instructions and data. Applications 220 may include, for example, benchmarking client application 222. In at least one of the various embodiments, benchmarking client application 222 may be used to

exchange communications to and from cost modeling platform server computer 107, including, but not limited to, queries, searches, API calls, or the like.

Other examples of application programs include calendars, search programs, email client applications, IM applications, SMS applications, Voice Over Internet Protocol (VOIP) applications, contact managers, task managers, transcoders, database programs, word processing programs, security applications, spreadsheet programs, games, search programs, and so forth.

Additionally, in one or more embodiments (not shown in the figures), client computer 200 may include an embedded logic hardware device instead of a CPU, such as, an Application Specific Integrated Circuit (ASIC), Field Programmable Gate Array (FPGA), Programmable Array Logic (PAL), or the like, or combination thereof. The embedded logic hardware device may directly execute its embedded logic to perform actions. Also, in one or more embodiments (not shown in the figures), the network computer may include a hardware microcontroller instead of a CPU. In at least one embodiment, the microcontroller may directly execute its own embedded logic to perform actions and access its own internal memory and its own external Input and Output Interfaces (e.g., hardware pins and/or wireless transceivers) to perform actions, such as System On a Chip (SOC), or the like.

### **Illustrative Network computer**

FIGURE 3 shows one embodiment of network computer 300 that may be included in a system implementing one or more embodiments of the described innovations. Network computer 300 may include many more or less components than those shown in FIGURE 3. However, the components shown are sufficient to disclose an illustrative embodiment for practicing these innovations. Network computer 300 may represent, for example, one embodiment of cost modeling platform server computer 107 of FIGURE 1.

As shown in the figure, network computer 300 includes a processor 302 in communication with a memory 304 via a bus 328. Network computer 300 also includes a power supply 330, network interface 332, audio interface 356, global positioning systems (GPS) receiver 362, display 350, keyboard 352, input/output interface 338, processor-readable stationary storage device 334, and processor-readable removable storage device 336. Power supply 330 provides power to network computer 300.

Network interface 332 includes circuitry for coupling network computer 300 to one or more networks, and is constructed for use with one or more communication protocols and

technologies including, but not limited to, protocols and technologies that implement any portion of the Open Systems Interconnection model (OSI model), global system for mobile communication (GSM), code division multiple access (CDMA), time division multiple access (TDMA), user datagram protocol (UDP), transmission control protocol/Internet protocol (TCP/IP), Short Message Service (SMS), Multimedia Messaging Service (MMS), general packet radio service (GPRS), WAP, ultra wide band (UWB), IEEE 802.16 Worldwide Interoperability for Microwave Access (WiMax), Session Initiation Protocol/Real-time Transport Protocol (SIP/RTP), or any of a variety of other wired and wireless communication protocols. Network interface 332 is sometimes known as a transceiver, transceiving device, or network interface card (NIC). Network computer 300 may optionally communicate with a base station (not shown), or directly with another computer.

Audio interface 356 is arranged to produce and receive audio signals such as the sound of a human voice. For example, audio interface 356 may be coupled to a speaker and microphone (not shown) to enable telecommunication with others and/or generate an audio acknowledgement for some action. A microphone in audio interface 356 can also be used for input to or control of network computer 300, for example, using voice recognition.

Display 350 may be a liquid crystal display (LCD), gas plasma, electronic ink, light emitting diode (LED), Organic LED (OLED) or any other type of light reflective or light transmissive display that can be used with a computer. Display 350 may be a handheld projector or pico projector capable of projecting an image on a wall or other object.

Network computer 300 may also comprise input/output interface 338 for communicating with external devices or computers not shown in FIGURE 3. Input/output interface 338 can utilize one or more wired or wireless communication technologies, such as USB™, Firewire™, WiFi, WiMax, Thunderbolt™, Infrared, Bluetooth™, Zigbee™, serial port, parallel port, and the like.

GPS transceiver 362 can determine the physical coordinates of network computer 300 on the surface of the Earth, which typically outputs a location as latitude and longitude values. GPS transceiver 362 can also employ other geo-positioning mechanisms, including, but not limited to, triangulation, assisted GPS (AGPS), Enhanced Observed Time Difference (E-OTD), Cell Identifier (CI), Service Area Identifier (SAI), Enhanced Timing Advance (ETA), Base Station Subsystem (BSS), or the like, to further determine the physical location

of network computer 300 on the surface of the Earth. It is understood that under different conditions, GPS transceiver 362 can determine a physical location for network computer 300.

Network computer 300 may also include sensors 364 for determining geolocation information (e.g., GPS), monitoring electrical power conditions (e.g., voltage sensors, current  
5 sensors, frequency sensors, and so on), monitoring weather (e.g., thermostats, barometers, anemometers, humidity detectors, precipitation scales, or the like), light monitoring, audio monitoring, motion sensors, or the like. Sensors 364 may be one or more hardware sensors that collect and/or measure data that is external to network computer 300

In at least one embodiment, however, network computer 300 may, through other  
10 components, provide other information that may be employed to determine a physical location of the client computer, including for example, a Media Access Control (MAC) address, IP address, and the like.

Human interface components can be physically separate from network computer 300, allowing for remote input and/or output to network computer 300. For example, information  
15 routed as described here through human interface components such as display 350 or keyboard 352 can instead be routed through the network interface 332 to appropriate human interface components located elsewhere on the network. Human interface components include any component that allows the computer to take input from, or send output to, a human user of a computer. Accordingly, pointing devices such as mice, styluses, track balls,  
20 or the like, may communicate through pointing device interface 358 to receive user input.

Memory 304 may include Random Access Memory (RAM), Read-Only Memory (ROM), and/or other types of memory. Memory 304 illustrates an example of computer-readable storage media (devices) for storage of information such as computer-readable instructions, data structures, program modules or other data. Memory 304 stores a basic  
25 input/output system (BIOS) 308 for controlling low-level operation of network computer 300. The memory also stores an operating system 306 for controlling the operation of network computer 300. It will be appreciated that this component may include a general-purpose operating system such as a version of UNIX, or LINUX™, or a specialized operating system such as Microsoft Corporation's Windows® operating system, or the Apple Corporation's  
30 IOS® operating system. The operating system may include, or interface with a Java virtual machine module that enables control of hardware components and/or operating system operations via Java application programs. Likewise, other runtime environments may be included.

Memory 304 may further include one or more data storage 310, which can be utilized by network computer 300 to store, among other things, applications 320 and/or other data. For example, data storage 310 may also be employed to store information that describes various capabilities of network computer 300. The information may then be provided to  
5 another device or computer based on any of a variety of events, including being sent as part of a header during a communication, sent upon request, or the like. Data storage 410 may also be employed to store social networking information including address books, buddy lists, aliases, user profile information, or the like. Data storage 310 may further include program code, data, algorithms, and the like, for use by a processor, such as processor 302 to execute  
10 and perform actions such as those actions described below. In one embodiment, at least some of data storage 310 might also be stored on another component of network computer 300, including, but not limited to, non-transitory media inside processor-readable removable storage device 336, processor-readable stationary storage device 334, or any other computer-readable storage device within network computer 300, or even external to network computer  
15 300. Data storage 310 may include, for example, model data 312, benchmark data 314, one or more datasets 316, or the like.

Applications 320 may include computer executable instructions which, when executed by network computer 300, transmit, receive, and/or otherwise process messages (e.g., SMS, Multimedia Messaging Service (MMS), Instant Message (IM), email, and/or  
20 other messages), audio, video, and enable telecommunication with another user of another mobile computer. Other examples of application programs include calendars, search programs, email client applications, IM applications, SMS applications, Voice Over Internet Protocol (VOIP) applications, contact managers, task managers, transcoders, database programs, word processing programs, security applications, spreadsheet programs, games,  
25 search programs, and so forth. Applications 320 may include cost modeling application 322, benchmarking application 324, web server application 326, other applications 328, or the like, that may perform actions further described below. In at least one of the various embodiments, one or more of the applications may be implemented as modules and/or components of another application. Further, in at least one of the various embodiments,  
30 applications may be implemented as operating system extensions, modules, plugins, or the like.

In at least one of the various embodiments, applications, such as, cost modeling application 322, web server application 326, other applications 328, or the like, may be arranged to employ geo-location information to select one or more localization features, such as, time zones, languages, currencies, calendar formatting, or the like. Localization features may be used in user-interfaces and well as internal processes and/or databases. In at least one of the various embodiments, geo-location information used for selecting localization information may be provided by GPS 362. Also, in some embodiments, geolocation information may include information providing using one or more geolocation protocol over the networks, such as, wireless network 108 and/or network 111.

Furthermore, in at least one of the various embodiments, cost modeling application 322 and/or benchmarking application 324 may be operative in a cloud-based computing environment. In at least one of the various embodiments, these applications, and others, that comprise the management platform may be executing within virtual machines and/or virtual servers that may be managed in a cloud-based based computing environment. In at least one of the various embodiments, in this context the applications may flow from one physical network computer within the cloud-based environment to another depending on performance and scaling considerations automatically managed by the cloud computing environment. Likewise, in at least one of the various embodiments, virtual machines and/or virtual servers dedicated to modeling application 322 and/or benchmarking application 324 may be provisioned and de-commissioned automatically.

Also, in at least one of the various embodiments, modeling application 322, benchmarking application 324, or the like, may located in virtual servers running in a cloud-based computing environment rather than being tied to one or more specific physical network computers.

Further, network computer 300 may also comprise hardware security module (HSM) 360 for providing additional tamper resistant safeguards for generating, storing and/or using security/cryptographic information such as, keys, digital certificates, passwords, passphrases, two-factor authentication information, or the like. In some embodiments, hardware security module may be employ to support one or more standard public key infrastructures (PKI), and may be employed to generate, manage, and/or store keys pairs, or the like. In some embodiments, HSM 360 may be arranged as a hardware card that may be installed in a network computer.

Additionally, in one or more embodiments (not shown in the figures), network computer 300 may include an embedded logic hardware device instead of a CPU, such as, an Application Specific Integrated Circuit (ASIC), Field Programmable Gate Array (FPGA), Programmable Array Logic (PAL), or the like, or combination thereof. The embedded logic hardware device may directly execute its embedded logic to perform actions. Also, in one or more embodiments (not shown in the figures), the network computer may include a hardware microcontroller instead of a CPU. In at least one embodiment, the microcontroller may directly execute its own embedded logic to perform actions and access its own internal memory and its own external Input and Output Interfaces (e.g., hardware pins and/or wireless transceivers) to perform actions, such as System On a Chip (SOC), or the like.

In at least one of the various embodiments, cost modeling application 322 may enable a user to generate budgets, allocation rules, recursive allocation, data model, cost models, total cost values for offerings, reports, or the like. Also in at least one of the various embodiments, modeling application 322 and/or benchmarking application 324 may employ processes, or parts of processes, similar to those described below.

### **Illustrative Logical System Architecture**

FIGURES 4-9 are presented to illustrate logical architectures at least one of the various embodiments for infrastructure benchmarking based on dynamic cost modeling.

FIGURE 4 shows, for at least one of the various embodiments model 400 that may be generated using cost modeling platform (CMP) server computer 107. In at least one of the various embodiments, in this example, model 400 represents a business system and starts with costs flowing from cost source 402, which may be a general ledger (GL) or other type of financial data. In this model, cost source 402 flows to labor 404, fixed asset 408, and to category mapping 406.

In at least one of the various embodiments, in model 400, labor 404 may be allocated 39% of costs, fixed assets 28% of costs, and the remaining 33% may be allocated to a mapping component. In at least one of the various embodiments, based on a selected category template, the costs allocated to mapping component 406 may be allocated to the various cost categories that may make up the business system being model.

In at least one of the various embodiments, model 400 shows that 100% of costs flow to IT resource tower 410. From IT resource tower 410 costs flow to services 412. Thus, for

this business system, model 400 shows that all the costs are allocated to producing the service offerings that the modeled business organization provides.

In at least one of the various embodiments, the budget and forecasting application may further enable users to gain additional understanding of how the costs from cost source  
5 402 flow through the entire system. Model 400 shows that storage device 414 is responsible for 2.5% for the costs coming from cost source 402. And, that 100% of the costs of storage device 414 flows into the general category of for storage, shown by the allocation trace that shows of 100% of flowing from the storage device 414 to storage component 416. Likewise, model 400 shows that physical servers 420 are burdened by 100% of the costs of storage 416.  
10 And, since the business organization modeled by model 400 includes hypervisors that run on physical servers, the costs associate with hypervisor 418 flow from physical server 420. In at least one of the various embodiments, cost for the server category, servers 422 is constituted out of physical servers 420 and hypervisors 418, thus the costs for server 422 flow from those components. Further the applications 424 component of the model may be burdened with  
15 100% of the costs associated with servers 422. Completing the loop, allocation rule 426 shows that service component 412 may be burdened with 100% of the costs associate with applications 424.

FIGURE 5 shows table 500 that may include information related to datasets that may be used by budget and forecasting platform 107 for generating business systems and data  
20 models. In at least one of the various embodiments, table 500 shows an example of source cost data in the form of a dataset of General Ledger (GL) accounting records that may be provided by at least one external data source. In at least one of the various embodiments, a dataset may have more or less columns and detail as shown in table 500. In at least one of the various embodiments, dataset information such as shown in table 500 may be provided in  
25 various well-known formats and structures. For example, table 500 may be provided as one or more, XML files, comma separated files, directly from database tables, or the like. Also, in at least one of the various embodiments, datasets may be provided in non-standard formats (e.g., proprietary) where custom scripts and applications may be employed to extract and/or parse values from the datasets.

30 In at least one of the various embodiments, other types of raw datasets may be provided by other external data sources to budget and forecasting platform 107. For example, datasets that include information about physical IT assets, fixed assets, software licenses,

employees, labor costs, insurance records, vendor costs, utility costs (electricity, water, sewer, broadband, natural gas, oil, or the like), consulting expenses, legal fees, or the like.

FIGURE 6 illustrates a portion of a logical architecture for model 600 that may be enabled by at least one of the various embodiments.

5 In at least one of the various embodiments, the model 600 may have five categories: GL 602, Servers 604, Storage 610, Support 614, and Business Units 620. In at least one of the various embodiments, each category contains a set of cost line items. For example, GL 602 includes cost line items for Payroll, Taxes and Rent. Likewise, Support 614 includes cost line items Email, Printers, Telephone, and Parking. In at least one of the various  
10 embodiments, each cost line item has one or more associated cost values. For example, Storage 610 has two cost line items, D1, and D2 (e.g., disk drives) having associated cost values of \$1000 each.

Allocation rules may be used to connect categories and/or objects of model 600. In at least one of the various embodiments, allocation rules may show how costs (money) flow  
15 between the categories. Resulting in a graph where the categories may be represented as nodes and the allocation rules may be represented as edges. In at least one of the various embodiments, generally a model may be represented by a directed acyclic graph but that is not a requirement for an operative model. In at least one of the various embodiments, a model graph may contain cycles that are resolved or estimated using mathematical techniques,  
20 including but not limited to Gaussian elimination, Cholesky decomposition or Newton's method.

In at least one of the various embodiments, allocation rule 606 represents a rule allocating 20% of the money in category GL 602 (source object) to Servers category 604 (target object). In this example, GL 602 includes a total of \$20,000, thus 20% of \$20,000  
25 (e.g., \$4,000) flows based on allocation rule 606 to Servers 604. Likewise, allocation rule 608 may allocate \$2,000 from GL 602 to Storage 610. The other allocation rules in model 600 allocate 100% of the money to the next category: allocation rule 612 directs 100% of the money (e.g., \$4,000) to flow to Support 614; allocation rule 618 directs 100% of the money in Support (e.g., \$4,000) to flow to Business Units 620; and allocation rule 616 directs 100%  
30 of the money from Storage 610 to flow to Business Units 620.

In at least one of the various embodiments, money that flows into the category may be allocated among the included cost line items. In at least one of the various embodiments, each category may have one or more rules that may describe the assignment ratios for how

the money in a category may be assigned to cost line items. For the categories 604, 610, 614, and 620, simple allocation rules assign the money in the category evenly among the cost line items comprising each category. GL 602 may have different assignment ratios, in this non-limiting example, the assignment ratio between the cost line items may be 50% to Payroll,

5 25% to Taxes, and 25% to Rent.

In at least one of the various embodiments, an assignment ratio may represent how the money in an actual budget may be assigned to the actual cost line items. In at least one of the various embodiments, rules may be applied that distribute the money based on formulas that may be defined by the users or administrators who designed the model. In at least one of the  
10 various embodiments, the assignment ratios and allocations may be modified as part of the modeling process.

The model 600 is a simplified model useful for facilitating discussion and understanding of the embodiments, since allocation rules for models of large commercial entities can be numerous and complex. However, model 600 is at least sufficient to enable  
15 one of ordinary skill in the art to practice what is claimed herein.

FIGURE 7 shows a portion of a logical architecture for benchmarking system 700 that performs actions in accordance with at least one of the various embodiments. In at least one of the various embodiments, benchmarking application 702 may be a representation of benchmarking application 324. In at least one of the various embodiments, benchmarking  
20 application 702 may be arranged to perform actions to generate benchmarking information based on one or more models, such as, cost model 600. In some embodiments, the models may be associated with financial activity/performance for one or more customers and/or entities. Typically, the customers may be separate enterprises, companies, departments within an enterprise, or the like. In some cases, customers may have more than one model for  
25 modeling different parts of their business or for performing different types of analyses, and so on.

Accordingly, in at least one of the various embodiments, system 700 may include, one or more (e.g., up to thousands) of community models, such as community model 704, community model 706, community model 708, and so on. Herein, community models are  
30 described separately to distinguish them for one or more particular customer models of interest. Accordingly, community models may be considered models for other customers. Thus, each community model may be considered a customer model or a community model depending on the context.

In at least one of the various embodiments, customers may have one or more raw customer models, such as, raw customer model 710 that models one or more systems (e.g., financial systems). In this context, the model is considered a raw model because the model may not map closely to a template model and/or a benchmark model. For example, the raw customer model may have sub-elements that are grouped and/or named using a taxonomy that is different than one used by another “standard” model, standard model 716.

In at least one of the various embodiments, the raw customer model referred to a model because of its structural/logical representation of the raw customer data. In some embodiments, raw customer models may be complex models generated by a cost modeling platform (CMP) or the like. In other cases, raw customer models may be as simple as a spreadsheet, an XML file, comma separated file, or the like. In some embodiments, the raw customer model represents a model generated and/or designed for another system. In other embodiments, the raw customer model may be a model generated by a CMP.

In at least one of the various embodiments, a benchmarking application, such as benchmarking application 702 may include a mapping engine that maps raw customer models to one or more standard models. For example, the mapping engine may be arranged to map a given raw customer model to one or more selected standard models, such as, standard model 716.

In at least one of the various embodiments, customer model 714 represents a model that is generated based on mapping the elements of a raw customer model to another model. In some embodiments, the customer model may be very similar to an original raw customer model. The variance depends on the deviation between the raw customer model and the standard model that provides the basis for the customer model. In some cases, the raw customer model may have been generated based on the same standard model associated with the customer model. Thus, in these cases, the raw customer model may be the same or similar as the customer model. In other cases, the raw customer model may deviate significantly from the standard model used for the customer model. In any case, the mapping engine included in the benchmarking application may be arranged to perform the necessary mappings and/or transformations to generate customer models, such as, customer model 714, for raw customer models, such as raw customer model 710, using a standard model, such as standard model 716.

In at least one of the various embodiments, the benchmarking application may employ various techniques to determine which of a plurality of standard models may be applicable to given raw customer model. In at least one of the various embodiments, the machine learning may be used to generate one or more classifiers for classifier a given raw customer model as being appropriate for a particular standard model. In other cases, rule based configuration information may employed select a standard model. Further, in some embodiments, a user may select a standard model. In some cases, the benchmarking application may employ some or all of the above techniques to recommend a standard model that may be confirmed/accepted by a user.

Similarly, in at least one of the various embodiments, benchmarking application 702 may employ various techniques for determining how elements in a raw customer model are mapped and/or transformed to elements that fit the selected standard model to generate the appropriate customer model.

In at least one of the various embodiments, information discovered by analyzing one or more of community models may employed for selecting an appropriate standard model and/or performing the mapping and transformation. For example, in at least one of the various embodiments, correlations between the raw community models and the community models may be used in part for determining which standard model to select and/or recommend for a new or incoming raw customer model. For example, an incoming raw customer model may be similar to one or more previously seen raw models (e.g., raw community models). Thus, standard models selected for these previously seen raw models may be also selected/recommend for the incoming raw customer models.

In at least one of the various embodiments, benchmarking application 702 may be arranged to process the data associated with one or more of the community models to produce one or more benchmark models. In at least one of the various embodiments, benchmark models may have the same or similar elements as the one or more standard models that were used to generate the community models. However, the data values (e.g., benchmark data) may be computed based on the actual values of the community data associated with the community models. In at least one of the various embodiments, benchmark values may be means, medians, histograms, curves, or the like, that are generated from community data that is associated with one or more community models.

In at least one of the various embodiments, information from a benchmark model, such as, benchmark model 712, may be compared with information included in a customer model. Accordingly, the customer may be enabled to compare the performance of the system represented by the customer model to one or more benchmark values. In at least one of the various embodiments, benchmark comparison information may be presented using various report formats and presentations. In some embodiments, reports may be interactive such that they enable a user to drill-down or otherwise navigate through the customer models.

Further, in at least one of the various embodiments, reports may be arranged to highlight or emphasize outlier values to bring them to attention of users. Also, in at least one of the various embodiments, comparison of benchmark information and/or a benchmark model to a customer model may trigger one or more notifications based on rules, configurations, or the like, or combination thereof. For example, in at least one of the various embodiments, if the variance between a data value in a customer model and its corresponding value in a benchmark model exceeds a defined threshold, a notification message may be generated to bring this condition to the attention of one or more users.

In at least one of the various embodiments, industry survey information, such as, industry survey information 718, may be employed in addition to using community model information to generate benchmark information. In at least one of the various embodiments, industry survey information may include curated information provided by one or more external/third-party sources of industry practices. Also, in at least one of the various embodiments, industry survey information 718 may include real-time and/or near real-time information streams that include information, such as, pricing data, energy costs, or the like.

In at least one of the various embodiments, models, model data (raw, customer, community, benchmark, or otherwise), industry survey information, or the like, may be stored locally, remotely, in a cloud environment, or the like, or combination thereof. Also, each model in system 700 may be associated with data representing the values for a particular enterprise or customer (shown in FIGURE 7 but not numbered explicitly).

FIGURE 8 shows a portion of a logical architecture for benchmarking system 800 that performs actions in accordance with at least one of the various embodiments. In at least one of the various embodiments, a benchmarking system, such as, system 800, may include a mapping engine, such as, mapping engine 802. In at least one of the various embodiments, mapping engine 802 may be incorporated in a benchmark application, such as, benchmarking

application 324. In at least one of the various embodiments, a benchmarking system may also include one or more raw customer models, such as, raw customer model 804; one or more standard models, such as, standard model 806; one or more customer models, such as, customer model 808; or the like.

5           In at least one of the various embodiments, as introduced above, mapping engine 802 may be arranged to map raw customer models onto customer models based on a standard model. In this example, raw customer model 804 is mapped to customer model 808 using standard model 806.

10           In this non-limiting example, raw customer model 804 includes two elements, elements 808 (e.g., Linux Distro 1 and Linux Distro 2 ) that represent two different types of operating system versions (e.g, Linux Distributions) that may be used in an enterprise. In this example, raw customer model 804 has one element per type of Linux distributions. Likewise, in this example, elements 810 represent elements for two different types of mainframe computers; elements 812 represents two type of Unix versions; element 814 represents a  
15           desktop operating system; element 816, represent desktop computers; and elements 818, represent two kinds of mobile computers (e.g., tablet computers and mobile phones

            In this example, standard model 806 is being used to generate customer model 808 from raw customer model 804. However, standard model 806 has similar but not exact elements as raw customer model 804. Accordingly, in at least one of the various  
20           embodiments, mapping engine 802 may be arranged to perform the mappings and transformations to generate customer model 808 from raw customer model 804 and standard model 806.

            In at least one of the various embodiments, one or more rules, scripts, custom circuitry (e.g., ASICs, FPGAs, or the like), configuration information, programs, or the like,  
25           or combination thereof, may be employed by mapping engine 802 to perform the necessary transformations to generate a customer model from a raw customer model. Also, in at least one of the various embodiments, a user graphical user interface may be generated and provided to enable a user to participate directly or indirectly in the mapping process. For example, in some cases, mapping engine 802 may be unable to automatically determine how  
30           one or elements of a raw customer model should be mapped into a customer model. Accordingly, in at least one of the various embodiments, a user interface may be generated that enable one or more users to perform some or all of the mapping process.

In at least one of the various embodiments, elements in a customer model may represent one or more elements from a corresponding raw customer model. In this example, element 820 represents all of the Linux servers for this customer. Accordingly, in this example, element 820 may be an aggregation of the values of elements 808 of raw customer model 804. Likewise, in this example, element 822 in customer model 808 may be an aggregation of elements 810; element 824 in customer model 808 may be an aggregation of elements 812; and element 830 in customer model 808 may be an aggregation of elements 818.

Further, in this example, mapping engine 802 has mapped element 814 in raw customer model 804 to element 826 in customer model 808; and mapped element 816 in raw customer model 804 to element 828 in customer model 808.

FIGURE 9 illustrates a logical representation of benchmark comparison table 900 that is in accordance with at least one of the various embodiments. In at least one of the various embodiments, a benchmark comparison, such as, table 900 may include various columns, such as, sub-towers 902, customer actuals 904, benchmark values 906, delta values 908, or the like. In at least one of the various embodiments, sub-towers 902 indicate the name or description of an element in a customer model. Customer actuals 904 represents the actual value associated with an element of a customer model. Benchmark values 906 represents the benchmark value that have been computed by the benchmarking application. And, delta 908, shows the difference between customer actuals 904 and benchmark values 906. In this example, delta 908 represents a percentage value computed by taking the difference of a customer actuals value and its corresponding benchmark value and dividing this difference by the corresponding benchmark value. One of ordinary skill in the art will appreciate the table 900 may be implemented using various data structures and/or formats, including, linked lists, arrays, database tables, comma-separated-value (CSV) files, or the like, or combination thereof. Also, a benchmark comparison table may have more or less columns than table 900. For example, a benchmark comparison table might not include a column such as delta 908. Likewise, for example, a benchmark comparison table may include additional columns, such as, standard deviations, min-max values for the benchmark values, mean of benchmark values, median of benchmark values, or the like.

In at least one of the various embodiments, a benchmark comparison table, such as table 900 may be employed to review how closely a customer model may be following a benchmark. For example, in this example, the customer actuals of money going toward Linux

is 300% more than the Linux benchmark value. See, table 900, row 910. Likewise, in this example, the customer is spending 80% less on Windows than the benchmark value. See, table 900, row 912. The benchmark comparison information enables a customer to quantitatively ascertain how their own enterprise is performing compared to other enterprises.

- 5 In this example, the customer may be alerted to the fact that they are spending far more on Linux servers than other customers.

In some embodiments, significant deviations from benchmark value may indicate that a benchmark model may need to be refined/revised, or the community models selected to contribute to a set of benchmark values may need to be modified.

## 10 **Generalized Benchmarking Operations**

FIGURES 10-13 represent the generalized infrastructure benchmarking based on dynamic cost modeling in accordance with at least one of the various embodiments. In at least one of the various embodiments, processes 1000, 1100, 1200, and 1300 described in conjunction with FIGURES 10-13 may be implemented by and/or executed on a single  
15 network computer, such as network computer 300 of FIGURE 3. In other embodiments, these processes or portions thereof may be implemented by and/or executed on a plurality of network computers, such as network computer 300 of FIGURE 3. However, embodiments are not so limited, and various combinations of network computers, client computers, virtual machines, or the like may be utilized. Further, in at least one of the various embodiments, the  
20 processes described in conjunction with FIGURES 10-13 may be operative in cost modeling and benchmarking architectures such as those described in conjunction with FIGURES 4-9.

FIGURE 10 illustrates a flowchart for process 1000 for infrastructure benchmarking based on dynamic cost modeling in accordance with at least one of the various embodiments. After a start block, at block 1002, in at least one of the various embodiments, benchmark data  
25 may be generated based on community models and associated community data.

At block 1004, in at least one of the various embodiments, raw customer models may be generated based on raw customer data. In at least one of the various embodiments, raw customer models may be generated by consuming data that includes information about a customer's enterprise and operations. In at least one of the various embodiments, if the  
30 benchmarking application is arranged to benchmark information technology spending, the customer data may be structured or unstructured information describing the customer's IT assets, expenses, or the like.

At block 1006, in at least one of the various embodiments, the benchmarking application may map the raw customer models to one or more standard models. In at least one of the various embodiments, the raw customer model may be arranged based on how the customer had organized its raw data. Accordingly, the raw customer model may not be organized to be conducive to making comparison with other customer models (e.g., community models). Thus, in at least one of the various embodiments, a mapping engine may execute mapping and transformations to generate a customer model that conforms to one or more standard models.

At block 1008, in at least one of the various embodiments, benchmarking comparison data may be generated to compare the customer model values and with benchmark values. See, FIGURE 13.

At block 1010, in at least one of the various embodiments, optionally, one or more benchmark comparison reports may be generated and/or provided to users. In at least one of the various embodiments, benchmark comparison reports may include tables, charts, graphs, or the like, or combination thereof, that describe how one or more customer models compare to one or more benchmark models.

In at least one of the various embodiments, the benchmark comparison reports may be interactive and presented in a graphical user-interface. In some embodiments, interactive reports may enable user to dynamically navigate to different elements of the model (drilldown) and/or view different visualizations (tables, charts, and so on) of the benchmark comparisons.

In at least one of the various embodiments, this block may be considered optional because in some embodiments reports may be selectively generated rather than generated all of the time.

At decision block 1012, in at least one of the various embodiments, if there may be new and/or updated raw customer data, control may loop back to block 1004; otherwise, control may flow to decision block 1014. In at least one of the various embodiments, customers may provide updated raw customer data to in response to changes, such as, changes to their enterprise, refinements, corrections, or the like, or combination thereof. In some embodiments, portions of the raw customer data may include streaming information that may be provided in real-time or near real-time (e.g., sales information, price changes, inventory/assets updates, or the like). In any event, in some embodiments,

changes/modifications to the raw customer data may be incorporated into the benchmarking system to update the customer model and the benchmark comparison information.

At decision block 1014, in at least one of the various embodiments, if there is new/updated community data, control may loop back to block 1002; otherwise, control may loop back to block 1010. In some embodiments, similar to as described above, changes to the relevant community data and/or community models may trigger the benchmarking information to be re-generated to account for the new/changed information. Otherwise, the system may wait for user requests to generate reports and/or if model/data changes occur.

FIGURE 11 illustrates an overview flowchart for process 1100 that generates a customer model for infrastructure benchmarking in accordance with at least one of the various embodiments. After a start block, at block 1102, in at least one of the various embodiments, raw customer model and raw customer data may be provided to a benchmarking application, such as, benchmarking application 324. As described above, raw customer models and their corresponding data may be provided in various formats. In simple cases, raw customer models may be spreadsheets, database table information, csv files, or the like. In other cases, raw customer models may be structured XML files, models from one or more cost modeling systems, or the like. In at least one of the various embodiments, the benchmarking application may be arranged to have one or more ingestion components to process/parse the provided raw customer information (models and data). In at least one of the various embodiments, a plugin architecture may be used to enable users to provide customized components that perform the actions required for ingesting a given raw customer model and its data.

At block 1104, in at least one of the various embodiments, a standard model may be selected for generating a customer model. In at least one of the various embodiments, there may be one or more standard models that are available. Accordingly, in at least one of the various embodiments, the benchmarking application may be arranged to compare the raw customer information with the available standard models to select a standard model. In some embodiments, users may provide input to influence which standard model is selected.

In at least one of the various embodiments, standard models may be used as templates to define how to map raw customer models to customer models.

At block 1106, in at least one of the various embodiments, a customer model may be generated based on the selected standard model, the raw customer model, and the raw customer data. In at least one of the various embodiments, a mapping engine components

included in a benchmarking application may be arranged to map one or more elements of the raw customer model to a customer model based on the selected standard model. In at least one of the various embodiments, the mapping engine may employ rules, configurations, machine learned classifiers, or the like, to determine how elements in the raw customer model are mapped and/or transformed into a customer model.

At block 1108, in at least one of the various embodiments, optionally, the customer model may be modified and/or revised based on information determined from the community models. In at least one of the various embodiments, the mapping of raw customer models into customer models may be influenced one or more community models.

In some embodiments, one or more community models may be explored to determine if they have elements in common with the customer model. Or, if their corresponding raw customer model is similar to the raw customer model of the customer model that is being processed. Accordingly, similarities may be exploited to provide information the mapping engine. In some embodiments, if the raw customer model of an incoming customer is similar or the same as raw customer models corresponding to one or more community models, the mapping engine may execute mapping actions similar to actions performed on the community models having the similar or same raw customer models.

For example, if a review of community models shows that raw customer model elements named “Linux Distro X” are mapped to “Server Operating Systems” rather than “Linux”, the mapping engine may deviate from an otherwise standard model accordingly. This is advantageous because it enables the mapping engine to learn from previous modifications to a standard model rather than requiring a user to manually customize the customer model.

At block 1110, in at least one of the various embodiments, the customer model may be further modified based on user input. In at least one of the various embodiments, at almost any time in the process of ingesting raw customer models, a user may intervene in the process. In some embodiments, a user may select the standard model user for generating the customer model. Also, a user may modify the mapping such that the generated customer model deviates from the standard model. Accordingly, in cases where the mapping engine has incorrectly mapped a raw customer model element to a customer model element, a user may be enabled to step in a correct the mistake.

In at least one of the various embodiments, a graphical user interface may be generated that enables the user manipulate the elements of a customer model and its mapping.

In some embodiments, the user may be enabled to modifying the mapping rules/configuration that are used by the mapping engine.

In at least one of the various embodiments, the changes made to a customer model by a user may be recorded by the mapping engine. Accordingly, the mapping engine may be arranged to automatically perform a user's actions if similar raw customer models are encountered and/or similar standard models are selected for generating customer models.

FIGURE 12 illustrates a flowchart for process 1200 for mapping raw customer model elements to a customer model in accordance with at least one of the various embodiments. After at start block, at block 1202, a mapping engine, such as, a mapping engine in benchmarking application 324, may begin iterating over the elements of a raw customer model.

In at least one of the various embodiments, a raw customer model may be a data structure that contains the raw customer data for a customer is being added to a modeling system, such as, cost modeling application 322. In at least one of the various embodiments, the raw customer model may comprised of structured of unstructured data that described the items being modeled. In some embodiments, a raw customer model may be akin to an inventory list of assets, quantities, values, and so on. In other cases, the raw customer model may be complicated tree or graph representation, including models such as those described in FIGURES 4-6.

Further, in at least one of the various embodiments, a raw customer model may be the product of one or more upstream processes employed to pre-process the information before providing it to a cost modeling application and/or benchmarking application.

At block 1204, in at least one of the various embodiments, the best matching element in the standard model may be determined. In at least one of the various embodiments, the mapping engine may be arranged to parse a raw customer model to determine its various sub-components (e.g., elements) and their relationships. In some cases, the relationships may be explicitly described in the raw customer model based on its structure. For example, if the raw customer model is defined by a SQL database dump file, there may be relationship information explicitly included in the raw customer model and its data that may be used by mapping engine.

In other cases, the mapping engine may infer relationships based shared fields in the elements, or the like. For example, if the raw customer model is based on multiple CSV or

text files, the mapping engine may employ pattern matching to identify potential relationships based on identifiers/key values that may be included in the text files.

In at least one of the various embodiments, mapping engine may employ one or more matching techniques to determine one or more best matches in the standard model. In some  
5       embodiments, there may be more the one element of the standard model that considered to  
the match the raw customer model element that is being considered. In such cases, in some  
embodiments, the standard model elements may organized into a list and ranked based on  
how close they match the current raw customer model. For example, a machine learning  
based classifier system used by the mapping engine may generate confidence scores that  
10       correspond to how close the raw customer elements match the standard model elements.  
Thus, in this example, if the scores for more than one standard customer model elements fall  
within a defined threshold value, those standard customer model elements may be selected.

At block 1206, in at least one of the various embodiments, the raw customer model  
element may be mapped to a customer model element based on the matched standard model  
15       element. In at least one of the various embodiments, if multiple standard model elements  
were matched, the mapping engine may be arranged to select one of the multiple matches  
using one or more techniques, such as, taking the highest scored element, presenting a  
selection list to a user, or the like.

At decision block 1208, in at least one of the various embodiments, if there are more  
20       elements that need mapping, control may loop back to block 1202; otherwise, control may be  
returned to a calling process. Note, in at least one of the various embodiments, there may be  
some elements in the raw customer model that may be determined to be unable to map using  
the selected standard model. In some embodiments, such elements may be mapped to an  
element in the customer model that captures unmapped elements. Thus, the customer model  
25       will still track those “unmapped” elements. Also, in some embodiments, a user may be  
enabled to go in a map the elements manually if desired.

FIGURE 13 illustrates a flowchart for process 1300 for generating benchmarking  
information in accordance with at least one of the various embodiments. After a start block,  
at block 1302, in at least one of the various embodiments, the type of benchmarking may be  
30       determined.

In at least one of the various embodiments, as discussed above, there may be different  
types of benchmarking processes directed for different purposes and/or industries. Also, in at  
least one of the various embodiments, benchmarking types may be focused on different

sources of information for comparison. In some embodiments, benchmarking may be based on comparisons with other parts within the same enterprise (e.g., different departments in the same company, or the like).

5 In at least one of the various embodiments, benchmarking may be targeted at different types of analysis, such as, financial, manufacturing, shipping, human resources, or the like, or combination thereof. Accordingly, in at least one of the various embodiments, a benchmarking application, such as, benchmarking application 324, may be arranged to support various types of benchmarking.

10 In at least one of the various embodiments, benchmarking may be based on one or more industry surveys that may be provided by third-party sources. Also, in at least one of the various embodiments, benchmarking may be based on other enterprises that have their model and/or data co-located in the same cloud environment. E.g., community models.

15 In at least one of the various embodiments, the benchmarking application may be arranged to use configuration information, templates, or the like, for defining the different type of benchmarking and the actions associated with each type of benchmarking. In some embodiments, the different types of benchmarking may be associated with one or more scripts, programs, custom circuitry (e.g., FPGAs, ASICs, PALs, and so on), modules, or the like, to perform some or all of the actions associated with a type of benchmarking.

20 At block 1304, in at least one of the various embodiments, the relevant community models may be determined, if any. In at least one of the various embodiments, some benchmarking types may incorporate information from one or more community models.

25 In at least one of the various embodiments, the benchmarking application may be arranged to identify one or more community models that have one or more similarities with the customer model that is the subject of the benchmarking. In some embodiments, the number of same types of elements included in the models may be used for determining similarity.

30 In at least one of the various embodiments, a similarity in the values of one or more model elements may also be examined when determining similarity. In some embodiments, two models may have similar model elements, (e.g., the same type of sub-towers) but with values that are much different. For example, two models may both share the same model elements of LAN, Linux, Mainframe, and so on. However, in this example, if one model A has a LAN value of \$1,000,000 and the model B has a LAN value of \$5,000, model B may

not be an appropriate model to use for generating benchmarking information for model A because of the wide disparity in values.

At block 1306, in at least one of the various embodiments, additional relevant information, if any, may be determined. In at least one of the various embodiments,  
5 information may be gathered from industry survey information, real-time signals, user input, or the like.

In at least one of the various embodiments, one or more sources of third-party industry survey may be available to provide information for determining benchmarking information. In at least one of the various embodiments, this benchmark values for model  
10 elements based on the survey information may be used as another way to compare and/or validate benchmark information generated from the community models. Likewise, the benchmarking information generated from the community models may be used to validate the survey information.

In at least one of the various embodiments, real-time signal information may be  
15 captured and included in the benchmark information. In at least one of the various embodiments, real-time signal information may include, geolocation information, weather information, power information (e.g., power quality, power outages, or the like), energy information (e.g., price/cost of energy), production information, sales feeds, price changes, market information streams, or the like, or combination thereof, that may be integrated into  
20 the benchmarking information. For example, the cost of operations for a data center may spike in quarter where there several power outages since the data center may have to use more expensive backup power than similarly arranged data centers located in regions that did not experience power outages. Likewise, in at least one of the various embodiments, differences in weather conditions may drive costs up or down between for model elements of  
25 different community/customer models.

Accordingly, the additional information may be incorporated into the benchmarking information as additional comparison information. For example, industry survey information may reported to compare with benchmark values. Also, in at least one of the various embodiments, the additional information may be used to generate weight values that may be  
30 applied to one or more generated benchmark values. For example, in at least one of the various embodiments, variances in values for similar model elements may be normalized based on variances of weather conditions, energy prices, local scarcity costs, or the like, or combination thereof.

In at least one of the various embodiments, the benchmarking application may be arranged to use configuration information, templates, or the like, for determining how to apply the additional information to the benchmarking information. In some embodiments, the different types of additional information may be associated with one or more scripts,  
5 programs, custom circuitry (e.g., FPGAs, ASICs, PALs, and so on), modules, or the like, to perform some or all of the actions associated with applying the additional information to the benchmarking information.

Further, in at least one of the various embodiments, a CMP server computer 324 is arranged to include one or more sensors for determining geolocation information (e.g., GPS),  
10 monitoring electrical power conditions (e.g., voltage sensors, current sensors, frequency sensors, and so on), monitoring weather (e.g., thermostats, barometers, anemometers, humidity detectors, precipitation scales, or the like), or the like.

For example, in at least one embodiment, geolocation information (such as latitude and longitude coordinates, or the like) is collected by a hardware GPS sensor and  
15 subsequently employed to modify the benchmarking information that is processed for one or more elements of the customer model. Also, in at least one embodiment, weather information (such as temperature, atmospheric pressure, wind speed, humidity, or the like) is collected by a hardware weather sensor and subsequently employed to modify the benchmarking information that is processed for one or more elements of the customer model.  
20 Additionally, in at least one embodiment, electrical power information (such as voltage, current, frequency, or the like) is collected by a hardware electrical power sensor and subsequently employed to modify the benchmarking information that is processed for one or more elements of the customer model.

At block 1308, in at least one of the various embodiments, the benchmarking  
25 application may iterate over the elements in the customer model. In at least one of the various embodiments, the benchmarking application may iterate of the elements comprising a customer model to process the benchmarking information for one or more elements of the customer model.

In at least one of the various embodiments, the benchmarking application may be  
30 arranged to automatically process all of the models that are based on the same standard model at the same time rather than generating benchmarking information for a single customer model.

In at least one of the various embodiments, the benchmarking application may be arranged to generate customized benchmarking for a particular customer model. Accordingly, in at least one of the various embodiments, a given customer model may have specialized/customized configuration associated with it that defines how information for its benchmarks should be generated.

At block 1310, in at least one of the various embodiments, benchmark information corresponding to the element of the customer model may be generated. As mentioned above, more than one model may be processed at the time. Likewise, the benchmarking application may be arranged to use one or concurrent methods to generate benchmarking information for multiple model elements and/or multiple customer models at the same time.

At decision block 1312, in at least one of the various embodiments, if there are more elements that need benchmark information, control may loop back to block 1308; otherwise, control may flow to block 1314.

At block 1314, in at least one of the various embodiments, optionally, a report may be generated by the benchmarking application. In at least one of the various embodiments, the benchmarking application may include benchmarking comparison information showing how the customer model compares to the benchmark model. One of ordinary skill in the art will appreciate that reports may be arranged in various formats. Some reports may compare more than one customer model with the benchmarking information; some reports may show comparisons of different portions of the benchmarking information; or the like. Further, a benchmark information report may be an internal data structure used for determining if notifications and/or alarms should be generated to bring attention to conditions of interest rather than displaying a report to a user. Next, control may be returned to a calling process.

It will be understood that each block of the flowchart illustration, and combinations of blocks in the flowchart illustration, can be implemented by computer program instructions. These program instructions may be provided to a processor to produce a machine, such that the instructions, which execute on the processor, create means for implementing the actions specified in the flowchart block or blocks. The computer program instructions may be executed by a processor to cause a series of operational steps to be performed by the processor to produce a computer-implemented process such that the instructions, which execute on the processor to provide steps for implementing the actions specified in the flowchart block or blocks. The computer program instructions may also cause at least some of the operational steps shown in the blocks of the flowchart to be performed in parallel.

These program instructions may be stored on some type of machine readable storage media, such as processor readable non-transitive storage media, or the like. Moreover, some of the steps may also be performed across more than one processor, such as might arise in a multi-processor computer system. In addition, one or more blocks or combinations of blocks in the flowchart illustration may also be performed concurrently with other blocks or combinations of blocks, or even in a different sequence than illustrated without departing from the scope or spirit of the invention.

Accordingly, blocks of the flowchart illustration support combinations of means for performing the specified actions, combinations of steps for performing the specified actions and program instruction means for performing the specified actions. It will also be understood that each block of the flowchart illustration, and combinations of blocks in the flowchart illustration, can be implemented by special purpose hardware-based systems, which perform the specified actions or steps, or combinations of special purpose hardware and computer instructions. The foregoing example should not be construed as limiting and/or exhaustive, but rather, an illustrative use case to show an implementation of at least one of the various embodiments of the invention.

Additionally, in one or more steps or blocks, may be implemented using embedded logic hardware, such as, an Application Specific Integrated Circuit (ASIC), Field Programmable Gate Array (FPGA), Programmable Array Logic (PAL), or the like, or combination thereof, instead of a computer program. The embedded logic hardware may directly execute embedded logic to perform actions some or all of the actions in the one or more steps or blocks. Also, in one or more embodiments (not shown in the figures), some or all of the actions of one or more of the steps or blocks may be performed by a hardware microcontroller instead of a CPU. In at least one embodiment, the microcontroller may directly execute its own embedded logic to perform actions and access its own internal memory and its own external Input and Output Interfaces (e.g., hardware pins and/or wireless transceivers) to perform actions, such as System On a Chip (SOC), or the like.

### **Illustrative Logical System Architecture for Benchmark Engines**

FIGURES 14-16 are presented to illustrate logical architectures and processes at least one of the various embodiments for infrastructure benchmarking based on dynamic cost modeling. In at least one of the various embodiments, models and benchmarking information may be used to generate a benchmark engine. In at least one of the various embodiments,

benchmark engines may be arranged to generate a benchmark value from provided customer information. For example, the customer may provide one or more entries based on their own infrastructure models to a benchmark engine that may generate a value that may be used for at least benchmarking.

5           FIGURE 14 illustrates a logical schematic of system 1400 for generating benchmark value for a customer. In at least one of the various embodiments, model item benchmark information 1402 may be provided to and/or processed by benchmark component generator 1404. In at least one of the various embodiments, benchmark component generator 1404 may be arranged to determine one or more benchmark components from model item benchmark  
10   information 1402.

          In at least one of the various embodiments, model item benchmark may be the collection of information from a variety of sources, as described for FIGURE 7. In at least one of the various embodiments, model item benchmark information may be computed from the community model data that represents actual costs associated with infrastructure items of  
15   other customers. Also, in at least one of the various embodiments, model item benchmarking information may also be based on industry survey information, such as, industry survey information 718.

          In at least one of the various embodiments, model items may be considered to be an item represented in a benchmark model. In at least one of the various embodiments,  
20   infrastructure models may have model items, such as, workstations, servers, storage, mainframes, mobile, or the like. (See, FIGURE 9). Accordingly, in at least one of the various embodiments, each customer model may be arranged to contain a value, such as, a cost value associated with each model item.

          In at least one of the various embodiments, the costs associated with a model item  
25   may have one or more separate drivers, such as, numbers of units, location of units, unit types, age of units, number of employees associated with the model items, and so on. In at least one of the various embodiments, the particular cost drivers may be determined based on the customer model and the benchmark models.

          In at least one of the various embodiments, each cost driver may contribute to the total  
30   cost of the model item. Accordingly, benchmark component generator 1404 may be arranged to generate benchmark components 1406 that correspond to one or more cost drivers of the model item. For example, benchmark component 1408 may represent how a single cost

driver impacts the overall costs of the model item. For example, benchmark component 1408 may represent an approximation modelling function such a cost/per unit over number of units.

In at least one of the various embodiments, benchmark components may be generated by applying one or more curve fitting algorithms using the benchmarking information of the community models to generate the models. In at least one of the various embodiments, each benchmark component may represent a different cost driver that may be associated with the model item.

In at least one of the various embodiments, benchmark components may have different impacts or correlations with the costs associated with a model item. In some embodiments, one or more benchmark components, representing one or more cost drivers, may have stronger correlation to the costs of a model item than others. Some benchmark components may have little correlation with actual costs while other benchmark components may have strong correlations.

In at least one of the various embodiments, benchmark component evaluator 1410 may be arranged to evaluate the significance of the various benchmark components. For example, in at least one of the various embodiments, benchmark components that have correlation strength value that exceeds a defined threshold value may be determined to be significant benchmark components.

In at least one of the various embodiments, additional features of a benchmark component may be employed to evaluate the component, such as, number of samples, variance of the values, excluded by configuration, or the like, or combination thereof.

For example, in at least one embodiment, geolocation information (such as latitude and longitude coordinates, or the like) is collected by a hardware GPS sensor and subsequently employed by a benchmark component evaluator to identify/determine and to evaluate the one or more benchmark components. Similarly, in at least one embodiment, weather information (such as temperature, atmospheric pressure, wind speed, humidity, or the like) is collected by a hardware weather sensor and subsequently employed by a benchmark component evaluator to identify/determine and to evaluate the one or more benchmark components. Additionally, in at least one embodiment, electrical power information (such as voltage, current, frequency, or the like) is collected by a hardware electrical power sensor and

subsequently employed by a benchmark component evaluator to identify/determine and to evaluate the one or more benchmark components.

In at least one of the various embodiments, benchmark component evaluator 1410 may be arranged to generate a benchmark engine, such as, benchmark engine 1412 that may be comprised of one or more benchmark components.

Accordingly, in at least one of the various embodiments, a customer may provide customer information to benchmark engine 1412 to generate a benchmark value, such as, benchmark value 1416. For example, in at least one of the various embodiments, benchmark engine 1412 may be a benchmark engine for benchmarking server costs. Accordingly, in at least one of the various embodiments, if a customer provides the proper inputs to the benchmark engine, it may generate benchmark value. In at least one of the various embodiments, the particular inputs that may be required and the particular benchmark value(s) that may be generated may vary depending on the particular benchmark components that comprise the benchmark engine.

In at least one of the various embodiments, model item information may include geolocation information provided from one or more geolocation devices such as GPS 362 and/or GPS 258. In some embodiments, geolocation information may be provided in real-time from active geolocation sensors. In other embodiments, geolocation information may be collected periodically from geolocation sensors.

FIGURE 15 illustrates graphical examples of benchmark components that may be generated by a benchmark component generator in accordance with at least one of the various embodiments. In at least one of the various embodiments, as described above, any given model item may have one or more cost drivers. The individual cost drivers may have more or less correlation, with the total costs corresponding to a model item. For example, component 1502 illustrates an example of the costs per unit associated with a model item decreasing as the cost driver increase. This example shows that the cost drivers corresponding to component 1502 have a significant negative correlation with costs for the item. A concrete example may be that the per unit total cost for server computers decreases and the number of servers under manage increases. This result reflects the cost benefit of economies of scale that may result from having more server computers under management.

In at least one of the various embodiments, the various components that contribute to the total costs of a model item may have more or less impact. For example, component 1504

illustrates a benchmark component that may have a different impact than component 1502. In this example, as the cost drivers increase that total costs are little affected. In at least one of the various embodiments, this may indicate that benchmark component 1504 may be evaluated as not being selected to be included in the benchmark engine because it does not provide significant benchmarking feedback since impact of the cost drive to the total costs of the model item may be below a define threshold.

In at least one of the various embodiments, some components may have step-wise impacts on the total costs of a model item. For example, benchmark component 1506 illustrates a benchmark component that is non-linearly negatively correlated to total costs.

In at least one of the various embodiments, a benchmark component evaluator, such as, benchmark component evaluator 1410 may be arranged compare the impact/contribution the various cost driver benchmark components have on the total costs of the model items. In some embodiments, one or more cost drivers may dominate the total costs or similarly one or more cost drivers may be closely correlated with the per unit total costs of a model item.

Likewise, one or more of the cost drivers may contribute little to the overall costs or may have little correlation to the total costs. Accordingly, in at least one of the various embodiments, the benchmark component evaluator may determine which components should be compiled into a benchmark engine.

In at least one of the various embodiments, benchmark components may include geolocation information provided from one or more geolocation devices such as GPS and/or GPS. In some embodiments, geolocation information may be provided in real-time from active geolocation sensors. In other embodiments, geolocation information may be collected periodically from geolocation sensors.

FIGURE 16 illustrates a logical representation of benchmark engine 1600 for server computers in accordance with at least one of the various embodiments. In this example, for at least one of the various embodiments, axis 1602 represents the unit cost of server computers; axis 1604 represents the number of units (the number of server computers in an enterprise); and curve 1606 represents the benchmark values for server computers. In at least one of the various embodiments, curve 1606 may be generated by  $y = F(x)$  where  $x$  represents the cost driver, the number of server computers in this example. In at least one of the various embodiments,  $F(x)$  may be comprised of one or more benchmark components. Accordingly, in some embodiments, the benchmark engine may produce benchmark values comprised of

one or more cost driver functions, such as,  $F(x) = a(x) + b(x) + \dots + z(x)$ . Likewise, in other embodiments, the benchmark engine may comprise one or more components functions for one or more cost drivers, such as,  $F(x, y, z) = a(x) + b(y) + c(z) \dots$ , or the like.

As described above, in at least one of the various embodiments, a benchmark component evaluator may be arranged to select the one or more benchmark components based on their individual impact to the cost value be modeled. Likewise, in at least one of the various embodiments, a benchmark component evaluator may be arranged to select/evaluate components based on other features, such as, amount of data, age of data, complexity of component, or the like, or combination thereof.

In at least one of the various embodiments, benchmark components and/or benchmark engine may be updated to reflect changes in the underlying community models and/or survey information. Accordingly, in at least one of the various embodiments, periodically benchmark components and/or benchmark engines may be re-generated to accommodate changes in the underlying data.

#### **Generalized Operations for Benchmarking Engines**

FIGURES 17-18 represent the generalized infrastructure benchmarking based on dynamic cost modeling in accordance with at least one of the various embodiments. In at least one of the various embodiments, processes 1700, and 1800 described in conjunction with FIGURES 17-18 may be implemented by and/or executed on a single network computer, such as network computer 300 of FIGURE 3. In other embodiments, these processes or portions thereof may be implemented by and/or executed on a plurality of network computers, such as network computer 300 of FIGURE 3. However, embodiments are not so limited, and various combinations of network computers, client computers, virtual machines, or the like may be utilized. Further, in at least one of the various embodiments, the processes described in conjunction with FIGURES 17-18 may be operative in cost modeling and benchmarking architectures such as those described in conjunction with FIGURES 4-9 and FIGURES 14-16.

FIGURE 17 illustrates a flowchart for process 1700 for generating benchmarking values using benchmarking engines in accordance with at least one of the various embodiments. After a start block, at block 1702, in at least one of the various embodiments, benchmark data may be generated based on community data and community models. In at least one of the various embodiments, benchmark data may be generated by from community

models that may be generated by raw model data supplied and/or obtained from customer infrastructure installations. In at least one of the various embodiments, benchmark data may be represented in data structures such as, tables, databases, or the like, organized using one or more standard models. (See, FIGURE 8.)

5           At block 1704, in at least one of the various embodiments, one or more benchmark components may be generated from the benchmark data. In at least one of the various embodiments, as described above, benchmark components may be used to represent the impact various cost drivers have on the model items comprising the standard models.

10           In at least one of the various embodiments, the total costs associated with a model item in a community model represents all of the costs that accrue for the model item. For example, if the model item is Linux Servers, the total costs may include hardware costs, energy costs, maintenance costs, employee costs (e.g., IT admin costs), software costs, or the like, that may be associated with a given community member (customer). Accordingly, in at least one of the various embodiments, community models each have cost values that are  
15           specific to their particular infrastructure. However, in at least one of the various embodiments, since each community model is based on a standard model, the impact of the various cost drivers may on total costs may be analyzed.

20           In at least one of the various embodiments, one or more benchmark components may be generated in part by using one or more curve fitting techniques to produce functions that represent how the total costs are related to a given cost driver. In some embodiments, the total costs may be considered as total costs per unit. This enables one or more benchmark component to represent how different values of a given cost driver affect the cost per unit of a model item, such as, server computers.

25           For example, in at least one embodiment, geolocation information (such as latitude and longitude coordinates, or the like) is collected by a hardware GPS sensor and subsequently employed by a benchmark component evaluator to identify/determine and to evaluate the one or more benchmark components. Similarly, in at least one embodiment, weather information (such as temperature, atmospheric pressure, wind speed, humidity, or the like) is collected by a hardware weather sensor and subsequently employed by a benchmark  
30           component evaluator to identify/determine and to evaluate the one or more benchmark components. Additionally, in at least one embodiment, electrical power information (such as voltage, current, frequency, or the like) is collected by a hardware electrical power sensor and

subsequently employed by a benchmark component evaluator to identify/determine and to evaluate the one or more benchmark components.

At block 1706, in at least one of the various embodiments, one or more benchmark engines may be generated based on the one or more benchmark components. In at least one of the various embodiments, as described above, benchmark engines may be comprised of or more benchmark components. In at least one of the various embodiments, the particular benchmark components that comprise a benchmark engine may be determined by an evaluation of the benchmark components. The selected benchmark components for a model item may be combined in to a benchmark engine for the model item.

At block 1708, in at least one of the various embodiments, customers may use the benchmark engines to generate benchmark values to use to compare against their own values. In at least one of the various embodiments, one or more user interfaces may be arranged to enable customers to use the benchmark values to evaluate the costs of their own infrastructure. In at least one of the various embodiments, the benchmark values provided to the users may be generated from the benchmark engines rather than the community models.

In at least one of the various embodiments, using the benchmark engines alleviates the need to use the actual benchmark data of the community members and/or survey data to produce the benchmark values. In at least one of the various embodiments, the benchmark engine produces benchmark values without using data supplied by other sources or customers. Accordingly, in at least one of the various embodiments, the benchmark engine produces benchmark values without accessing benchmark data supplied and/or owned by other sources or customers.

At decision block 1710, in at least one of the various embodiments, if the benchmark engines are ready to be updated, control may loop back to block 170; otherwise control may continue to decision block 1712. In at least one of the various embodiments, as benchmark data associated with the community models and/or survey data changes overtime the benchmark engines may need to be updated. In at least one of the various embodiments, the updating may occur periodically based on a configured time period, such as, once a month, once a week, or the like.

In other embodiments, the updating may be triggered based on the amount of changes to the benchmark data that are recorded. Accordingly, one or more threshold values may be defined that if exceeded may trigger an update to occur. For example, in at least one of the various embodiments, an update may be triggered if the number of community models

increases (or decreases) such that a threshold value is exceeded. Likewise, for example, in at least one of the various embodiments, changes in data such as model item prices, costs, sales revenue, or the like, or combination thereof, may trigger an update.

At decision block 1712, in at least one of the various embodiments, if the customer(s)  
5 are not finished generating/viewing benchmark values, control may loop back to block 1708; otherwise, control may be returned to a calling process. In at least one of the various embodiments, generating benchmark values may be activated by user via of a user-interface. However, in other embodiments, the benchmark values may be provided by an application and/or service that provides the values to another application. For example, in some  
10 embodiments, the benchmark engines may be arranged to be accessible by various methods, such as, APIs, REST APIs, or the like, or combination thereof.

FIGURE 18 illustrates a flowchart for process 1800 for generating benchmarking values using benchmarking engines in accordance with at least one of the various embodiments. After a start block, at block 1802, in at least one of the various embodiments,  
15 benchmark model items may be determined for generating one or more benchmark components. As described above, benchmark components may be generated for selected model items. In some embodiments, the model items may be selected based on configuration information. For example, in some embodiments, the model items may selected based on their commonality across the community, proportion of cost (e.g., the model items having the  
20 top 'n' biggest impacts on budgets may be selected), availability of relevant benchmark data, or the like, or combination thereof.

At block 1804, in at least one of the various embodiments, one or more benchmark components for the selected model items may be generated. In at least one of the various embodiments, the particular actions employed to generate the components may be determined  
25 based on configuration information. Also, in some embodiments, the particular selection of cost drivers, units, and so on, may vary depending on the model items and may be determined based on configuration information.

In at least one of the various embodiments, linear regression may be employed to determine the relationship between model item costs and the cost drivers. Also, goodness of  
30 fit of benchmark component curve may be evaluated using least squares regression.

In at least one of the various embodiments, other well-known techniques, such as, Bayesian multivariate linear regression may be used for modeling/curve fitting. Also, in at least one of the various embodiments, configuration settings, or other rules, may be in place

to filter and/or establish floors and/or ceilings for the base values based on a priori knowledge of the model items and their relevant cost drivers.

For example, in at least one of the various embodiments, for a model item such as desktop computers, configuration information may indicate that the benchmark components are to be generated in terms of cost per unit. In this example, benchmark components may be generated for computing a desktop computer unit cost vs number of desktop computers; unit cost vs. number of employees, unit cost vs. number of vehicles, unit cost vs. gross sales, or the like.

Likewise, for example, a model item such as environmental expenses (e.g., heating and air-conditioning) benchmark components represent total cost vs. number of employees; total cost vs. number of computer; or the like.

Accordingly, in at least one of the various embodiments, configuration information may be arranged to define the benchmarking units for each model item and/or class/type of model item. Also, in at least one of the various embodiments, customer may customize/configuration the particular model items and/or the unit they would to include as benchmark components. In at least one of the various embodiments, such customization may be provided using a graphical user-interfaces, configuration files, or the like.

At block 1806, in at least one of the various embodiments, optionally, the one or more benchmark components may be further modified based on configuration information and/or user provided customization. For example, one or more benchmark components may be modified because there may be prior knowledge that the benchmark data used to generate them is biased. In at least one of the various embodiments, third party data sources may be consulted to determine if a benchmark component may be based. Also, in at least one of the various embodiments, where community model sample sizes are below a defined threshold, biasing based on one or more external data sources and/or datasets, such as, industry/category standards may be employed to account for biases, such as, biases that may be introduced by a small sample size, or the like.

At block 1808, in at least one of the various embodiments, the one or more generated benchmark components may be evaluated to determine if they should be incorporated into the benchmark engine for the model item. As discussed above, the various benchmark components may have different impacts on the values (e.g., total unit cost) that are being benchmarked. Accordingly, in at least one of the various embodiments, one or more of the

less impactful benchmark components may be determined to be excluded from the benchmark engine.

In at least one of the various embodiments, results of best fit analysis, such as, correlation coefficients, goodness of fit, coefficient of determination, chi-squared test, or the like, may be employed to evaluate one or more benchmark components. In at least one of the various embodiments, benchmark components may be rank ordered based on the evaluation. Also, in at least one of the various embodiments, benchmark components may be excluded or included based on exceeding one or more defined threshold values. For example, a predefined threshold value of 0.5 may be defined such that benchmark components having correlation coefficients with a magnitude of less than 0.5 may be excluded. Likewise, configuration information may define a threshold value of 0.9 may be defined such that benchmark components having correlation coefficients with a magnitude of 0.9 or greater may be automatically included in a benchmark engine.

At decision block 1810, in at least one of the various embodiments, if there are more benchmark components to process, control may loop back to block 1804; otherwise, control may flow to block 1812.

At block 1812, in at least one of the various embodiments, one or more of the benchmark components may be incorporated in a benchmark engine. In at least one of the various embodiments, the particular benchmark components may be selected based on the results of the evaluations described in block 1806. In at least one of the various embodiments, only the top benchmark component may be incorporated in the benchmark engine. In at least one of the various embodiments, there may be a defined “top” number of benchmark components that may be selected. Still in other embodiments, the configuration information, such as rule-based policies may be employed to determine the one or more benchmark components that may be included in a benchmark engine. For example, configuration information may be defined such that benchmark components having correlation coefficients with a magnitude of 0.9 or greater may be automatically included in a benchmark engine. Next, control may be returned to a calling process.

It will be understood that each block of the flowchart the illustrations, and combinations of blocks in the flowchart illustrations, can be implemented by computer program instructions. These program instructions may be provided to a processor to produce a machine, such that the instructions, which execute on the processor, create means for implementing the actions specified in the flowchart block or blocks. The computer program

instructions may be executed by a processor to cause a series of operational steps to be performed by the processor to produce a computer-implemented process such that the instructions, which execute on the processor to provide steps for implementing the actions specified in the flowchart block or blocks. The computer program instructions may also  
5 cause at least some of the operational steps shown in the blocks of the flowcharts to be performed in parallel. Moreover, some of the steps may also be performed across more than one processor, such as might arise in a multi-processor computer system. In addition, one or more blocks or combinations of blocks in the flowchart illustration may also be performed concurrently with other blocks or combinations of blocks, or even in a different sequence  
10 than illustrated without departing from the scope or spirit of the invention.

Additionally, in one or more steps or blocks, may be implemented using embedded logic hardware, such as, an Application Specific Integrated Circuit (ASIC), Field Programmable Gate Array (FPGA), Programmable Array Logic (PAL), or the like, or combination thereof, instead of a computer program. The embedded logic hardware may  
15 directly execute embedded logic to perform actions some or all of the actions in the one or more steps or blocks. Also, in one or more embodiments (not shown in the figures), some or all of the actions of one or more of the steps or blocks may be performed by a hardware microcontroller instead of a CPU. In at least one embodiment, the microcontroller may directly execute its own embedded logic to perform actions and access its own internal  
20 memory and its own external Input and Output Interfaces (e.g., hardware pins and/or wireless transceivers) to perform actions, such as System On a Chip (SOC), or the like.

### **Illustrative Use Cases**

FIGURES 19-23 illustrate non-limiting examples of user interfaces for generalized infrastructure benchmarking based on dynamic cost modeling in accordance with at least one  
25 of the various embodiments. One of ordinary skill in the art will appreciate that other user interface may be used without departing for the scope of the innovations described herein. Further, the below described user interfaces are sufficient of ordinary skill in the art to understand and practice the innovations described herein.

FIGURE 19 illustrates user interface 1900 for displaying benchmarking information  
30 in accordance with at least one of the various embodiments. In this example, user interface 1900 includes graphical and tabular representation of benchmarking information that may be generated by a benchmarking application, such as, benchmarking application 324. One

ordinary skill in the art will appreciate that benchmarking information may be displayed to users using various user-interface and/or reporting formats. In this example, a bar chart may be arranged to visually depict the benchmarking values. Accordingly, column 1902 may represent a benchmark value and column 1904 may represent the actual value that is compared to the benchmark. Likewise, in this example, the benchmarking information may be presented in a tabular format. Here, table column 1906 represents items or aggregated items in the model; table column 1908 represent the benchmark value and table column 1910 represents the actual value.

FIGURE 20 illustrates user interface 2000 for displaying benchmarking information in accordance with at least one of the various embodiments. In this example, user interface 2000 includes graphical and tabular representation of benchmarking information that may be generated by a benchmarking application, such as, benchmarking application 324. In this example, user interface 2000 is a drill-down view that shows benchmarking information about sub-towers of a cost model.

Accordingly, in at least one of the various embodiments, value 2002 may summarize the benchmark value for the sub-tower (e.g., component of the model); value 2004 may summarize the actual value for the sub-tower; and value 2006 may summarize the difference between the benchmark and the actuals. Also, in at least one of the various embodiments, a bar chart may be arranged to visually depict the benchmarking values. Accordingly, column 2008 may represent a benchmark value and column 2010 may represent the actual value that is compared to the benchmark.

FIGURE 21 illustrates user interface 2100 for displaying benchmarking information in accordance with at least one of the various embodiments. In this example, user interface 2100 includes graphical and tabular representation of benchmarking information that may be generated by a benchmarking application, such as, benchmarking application 324. In this example, the information may be related to a single element of a sub-tower. Tab 2104 may indicate the current item in the sub-tower being reviewed (e.g., Windows). Tab 2102 may enable a user to navigate to a higher level view the represents the entire sub-tower (e.g., FIGURE 2000).

Also, in at least one of the various embodiments, a bar chart may be arranged to visually depict the benchmarking values. Accordingly, column 2106 may represent a

benchmark value and column 2108 may represent the actual value that is compared to the benchmark.

FIGURE 22 illustrates user interface 2200 for viewing/generating mapping rules for mapping raw customer models to customer models in accordance with at least one of the various embodiments. In this example, user interface 2200 includes tabular representation of model elements that may be mapped using a benchmarking application, such as, benchmarking application 324.

FIGURE 23 illustrates user interface 2300 for reviewing mapped models in accordance with at least one of the various embodiments. In this example, user interface 2300 includes tabular representation of model elements that may be have been mapped using a benchmarking application, such as, benchmarking application 324.

## CLAIMS

What is claimed as new and desired to be protected by Letters Patent of the United States is:

1. A method for managing resource allocation using a network computer that performs actions comprising:

generating one or more community models based on raw customer model data, wherein the raw customer model data is mapped to the one or more community models based on one or more rules for mapping the raw customer model data to one or more standard models;

determining a plurality of model items based on the one or more community models;

generating one or more benchmark components that correspond to one or more of the plurality of model items based on benchmark data included in the one or more community models;

ranking the one or more benchmark components for each of the plurality of model items based on an evaluation of each benchmark component's contribution to a total cost value for a corresponding model item;

generating one or more benchmark engines that correspond to one or more of the plurality of model items based on a combination of benchmark components, wherein the combination of the benchmark components is selected based on a ranking;

when an update to a value for raw customer model data exceeds a threshold, re-generating the one or more benchmark engines based on the updated value for the raw customer model data; and

when an actual data value is provided for the one or more of the plurality of model items, using the one or more benchmark engines to generate one or more benchmark values.

2. The method of Claim 1, further comprising, when a change in a number of community models exceeds a threshold value, updating the one or more benchmarking engines based on the change to the number of community models.

3. The method of Claim 1, further comprising, modifying the benchmark data included in the one or more community models based on real-time information obtained from one or more of geolocation sensors, weather information sensors, or electrical power sensors.
4. The method of Claim 1, wherein ranking the one or more benchmark components, further comprises, employing best fit analysis, including one or more, correlation coefficients, goodness of fit, coefficient of determination, or chi-squared test, to evaluate the one or more benchmark components.
5. The method of Claim 1, wherein generating the one or more benchmark components, further comprises, modifying the one or more benchmark components to account for one or more biases, wherein the modifying is based on one or more external data sources.
6. The method of Claim 1, wherein including the benchmarking data in the one or more community models, further comprises:
  - determining additional relevant information from one or more external data sources; and
  - modifying the benchmarking data based on the additional relevant information.
7. A system for managing resource allocation, comprising:
  - a network computer, comprising:
    - a transceiver that communicates over the network;
    - a memory that stores at least instructions; and
    - a processor device that executes instructions that perform actions,including:
  - generating one or more community models based on raw customer model data, wherein the raw customer model data is mapped to the one or more community models based on one or more rules for mapping the raw customer model data to one or more standard models;

determining a plurality of model items based on the one or more community models;

generating one or more benchmark components that correspond to one or more of the plurality of model items based on benchmark data included in the one or more community models;

ranking the one or more benchmark components for each of the plurality of model items based on an evaluation of each benchmark component's contribution to a total cost value for a corresponding model item;

generating one or more benchmark engines that correspond to one or more of the plurality of model items based on a combination of benchmark components, wherein the combination of the benchmark components is selected based on a ranking;

when an update to a value for raw customer model data exceeds a threshold, re-generating the one or more benchmark engines based on the updated value for the raw customer model data; and

when an actual data value is provided for the one or more of the plurality of model items, using the one or more benchmark engines to generate one or more benchmark values; and

a client computer, comprising:

a transceiver that communicates over the network;

a memory that stores at least instructions; and

a processor device that executes instructions that perform actions, including:

displaying the one or more benchmark values.

8. The system of Claim 7, wherein the network computer processor device executes instructions that perform actions, further comprising, when a change in a number of community models exceeds a threshold value, updating the one or more benchmarking engines based on the change to the number of community models.

9. The system of Claim 7, wherein the network computer processor device executes instructions that perform actions, further comprising, modifying the benchmark data included in

the one or more community models based on real-time information obtained from one or more of geolocation sensors, weather information sensors, or electrical power sensors.

10. The system of Claim 7, wherein ranking the one or more benchmark components. further comprises, employing best fit analysis, including one or more, correlation coefficients, goodness of fit, coefficient of determination, or chi-squared test, to evaluate the one or more benchmark components.

11. The system of Claim 7, wherein generating the one or more benchmark components, further comprises, modifying the one or more benchmark components to account for one or more biases, wherein the modifying is based on one or more external data sources.

12. The system of Claim 7, wherein including the benchmarking data in the one or more community models, further comprises:

determining additional relevant information from one or more external data sources; and

modifying the benchmarking data based on the additional relevant information.

13. A processor readable non-transitory storage media that includes instructions for managing resource allocation, wherein execution of the instructions by a hardware processor performs actions, comprising:

generating one or more community models based on raw customer model data, wherein the raw customer model data is mapped to the one or more community models based on one or more rules for mapping the raw customer model data to one or more standard models;

determining a plurality of model items based on the one or more community models;

generating one or more benchmark components that correspond to one or more of the plurality of model items based on benchmark data included in the one or more community models;

ranking the one or more benchmark components for each of the plurality of model items based on an evaluation of each benchmark component's contribution to a total cost value for a corresponding model item;

generating one or more benchmark engines that correspond to one or more of the plurality of model items based on a combination of benchmark components, wherein the combination of the benchmark components is selected based on a ranking;

when an update to a value for raw customer model data exceeds a threshold, re-generating the one or more benchmark engines based on the updated value for the raw customer model data; and

when an actual data value is provided for the one or more of the plurality of model items, using the one or more benchmark engines to generate one or more benchmark values.

14. The media of Claim 13, further comprising, when a change in a number of community models exceeds a threshold value, updating the one or more benchmarking engines based on the change to the number of community models.

15. The media of Claim 13, further comprising, modifying the benchmark data included in the one or more community models based on real-time information obtained from one or more of geolocation sensors, weather information sensors, or electrical power sensors.

16. The media of Claim 13, wherein ranking the one or more benchmark components. further comprises, employing best fit analysis, including one or more, correlation coefficients, goodness of fit, coefficient of determination, or chi-squared test, to evaluate the one or more benchmark components.

17. The media of Claim 13, wherein generating the one or more benchmark components, further comprises, modifying the one or more benchmark components to account for one or more biases, wherein the modifying is based on one or more external data sources.

18. The media of Claim 13, wherein including the benchmarking data in the one or more community models, further comprises:

- determining additional relevant information from one or more external data sources; and
- modifying the benchmarking data based on the additional relevant information.

19. A network computer for managing resource allocation, comprising:

- a transceiver that communicates over the network;
- a memory that stores at least instructions; and
- a processor device that executes instructions that perform actions, including:
  - generating one or more community models based on raw customer model data, wherein the raw customer model data is mapped to the one or more community models based on one or more rules for mapping the raw customer model data to one or more standard models;
  - determining a plurality of model items based on the one or more community models;
  - generating one or more benchmark components that correspond to one or more of the plurality of model items based on benchmark data included in the one or more community models;
  - ranking the one or more benchmark components for each of the plurality of model items based on an evaluation of each benchmark component's contribution to a total cost value for a corresponding model item;
  - generating one or more benchmark engines that correspond to one or more of the plurality of model items based on a combination of benchmark components, wherein the combination of the benchmark components is selected based on a ranking;
  - when an update to a value for raw customer model data exceeds a threshold, re-generating the one or more benchmark engines based on the updated value for the raw customer model data; and

when an actual data value is provided for the one or more of the plurality of model items, using the one or more benchmark engines to generate one or more benchmark values.

20. The network computer of Claim 19, wherein the processor device executes instructions that perform actions, further comprising, when a change in a number of community models exceeds a threshold value, updating the one or more benchmarking engines based on the change to the number of community models.

21. The network computer of Claim 19, wherein the processor device executes instructions that perform actions, further comprising, modifying the benchmark data included in the one or more community models based on real-time information obtained from one or more of geolocation sensors, weather information sensors, or electrical power sensors.

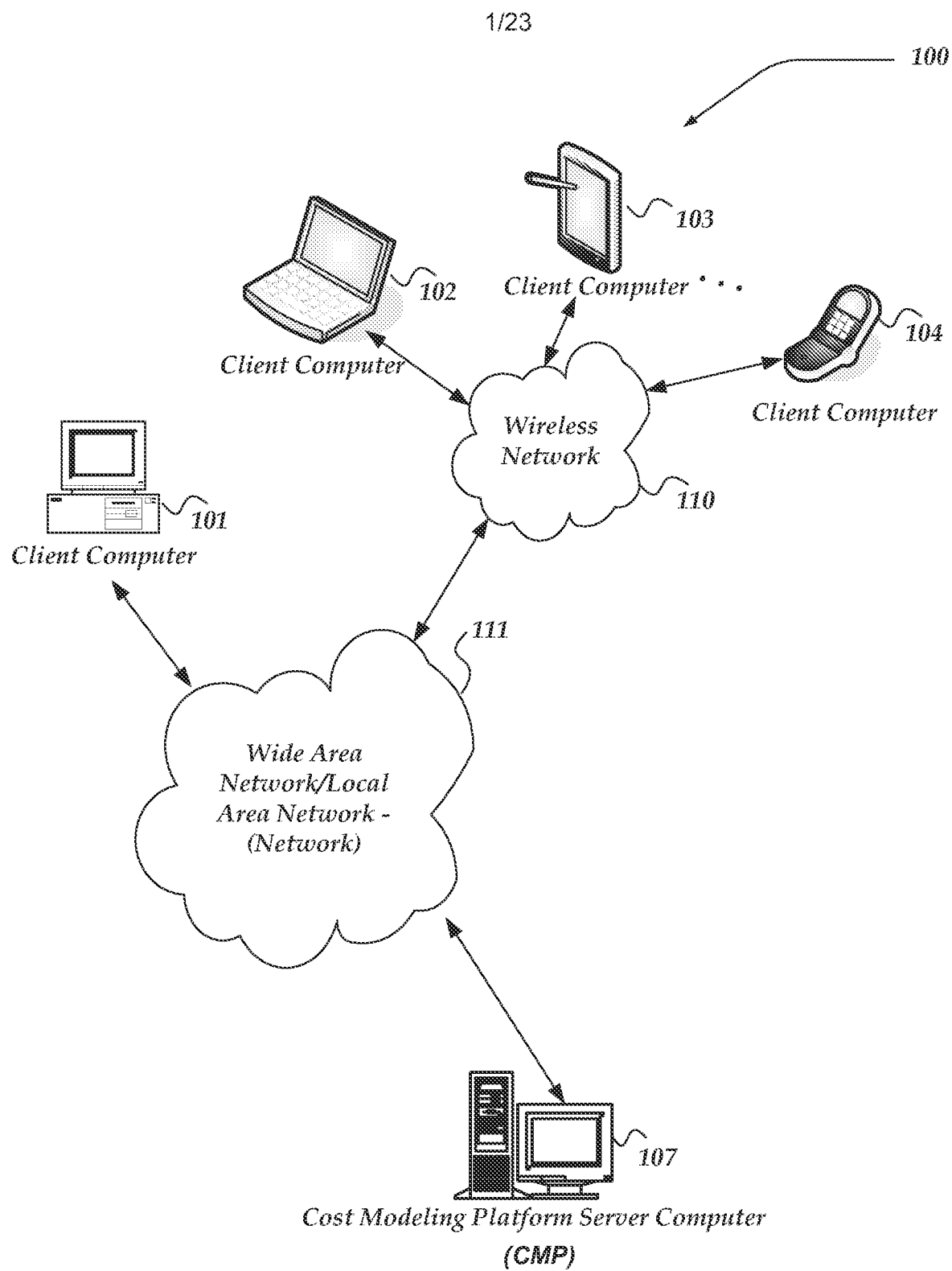
22. The network computer of Claim 19, wherein ranking the one or more benchmark components, further comprises, employing best fit analysis, including one or more, correlation coefficients, goodness of fit, coefficient of determination, or chi-squared test, to evaluate the one or more benchmark components.

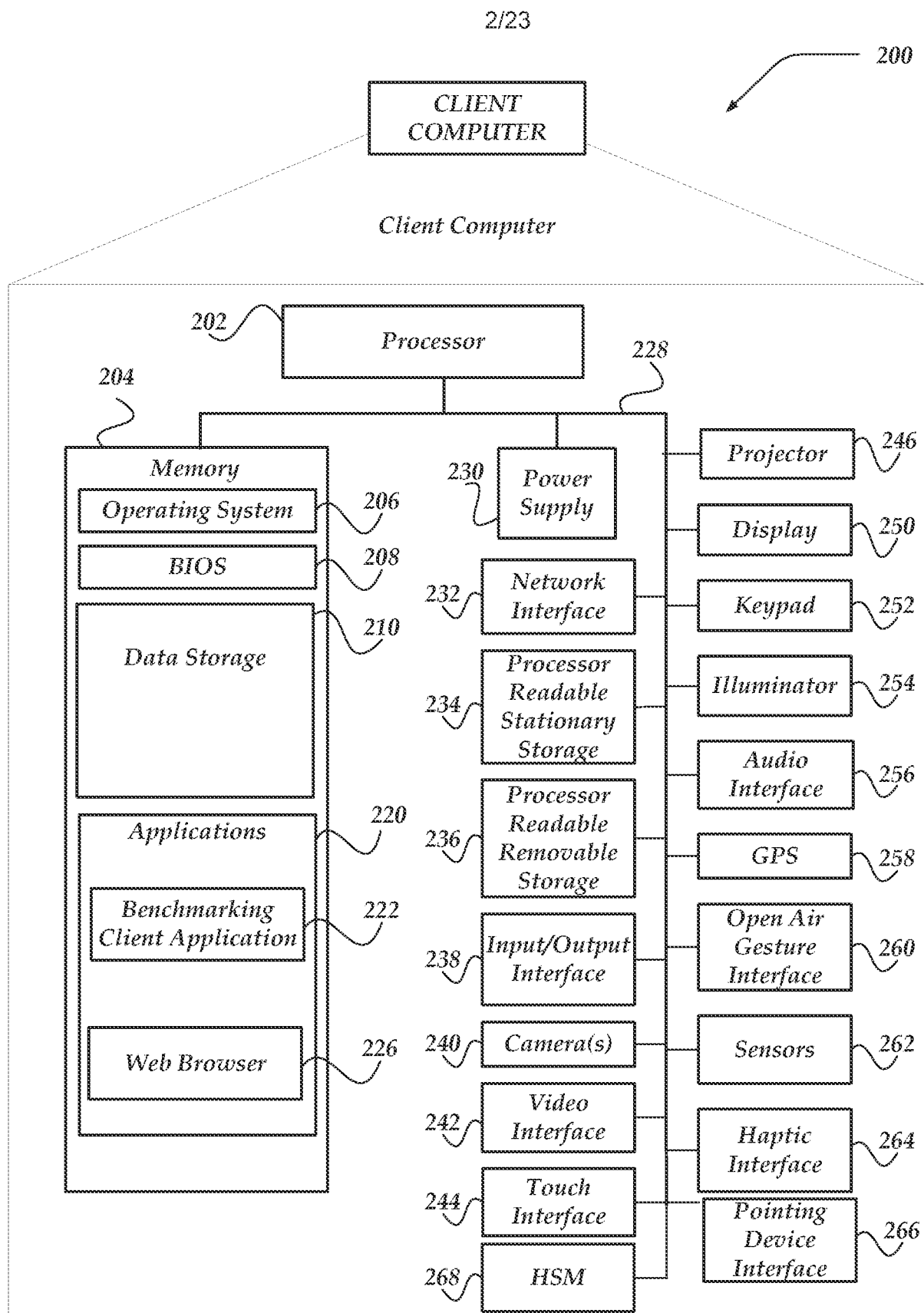
23. The network computer of Claim 19, wherein generating the one or more benchmark components, further comprises, modifying the one or more benchmark components to account for one or more biases, wherein the modifying is based on one or more external data sources.

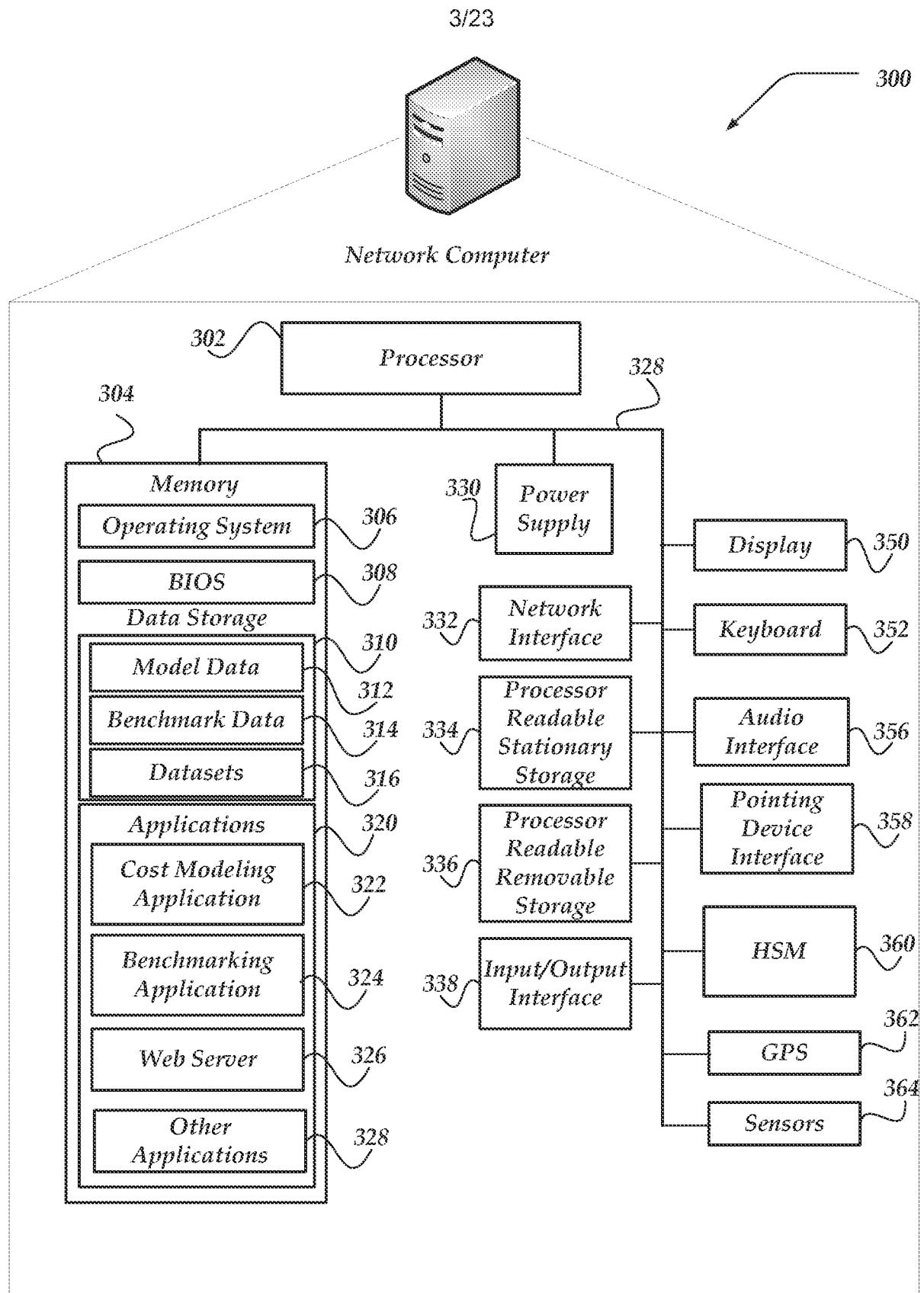
24. The network computer of Claim 19, wherein including the benchmarking data in the one or more community models, further comprises:

determining additional relevant information from one or more external data sources; and

modifying the benchmarking data based on the additional relevant information.

**FIG. 1**

**Fig. 2**

**Fig. 3**

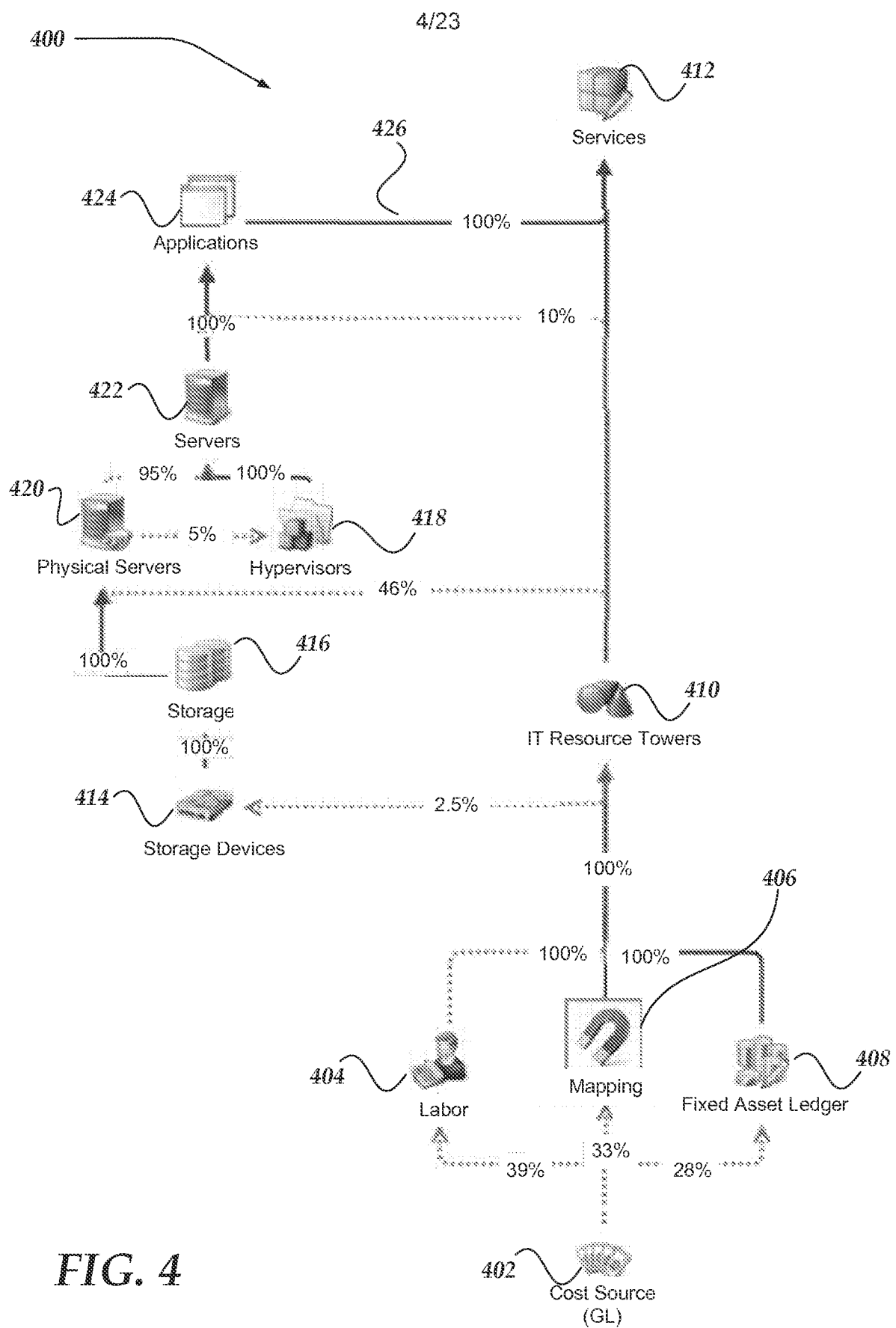


FIG. 4

5/23

500 

| GL Actuals.ID                            | Cost      |
|------------------------------------------|-----------|
| 301, Payroll (IT OPERATIONS)             | \$237,000 |
| 330, Employee Benefits (IT OPERATIONS)   | \$49,000  |
| 320, Payroll Taxes (IT OPERATIONS)       | \$4,100   |
| 760, Repairs (IT OPERATIONS)             | \$18,000  |
| 759, Rent Equipment (IT OPERATIONS)      | \$18,000  |
| 711, Fees and Consulting (IT OPERATIONS) | \$7,600   |
| 650, Training (IT OPERATIONS)            | \$6,200   |
| 777, Miscellaneous Expense               | \$1,800   |

**FIG. 5**

6/23

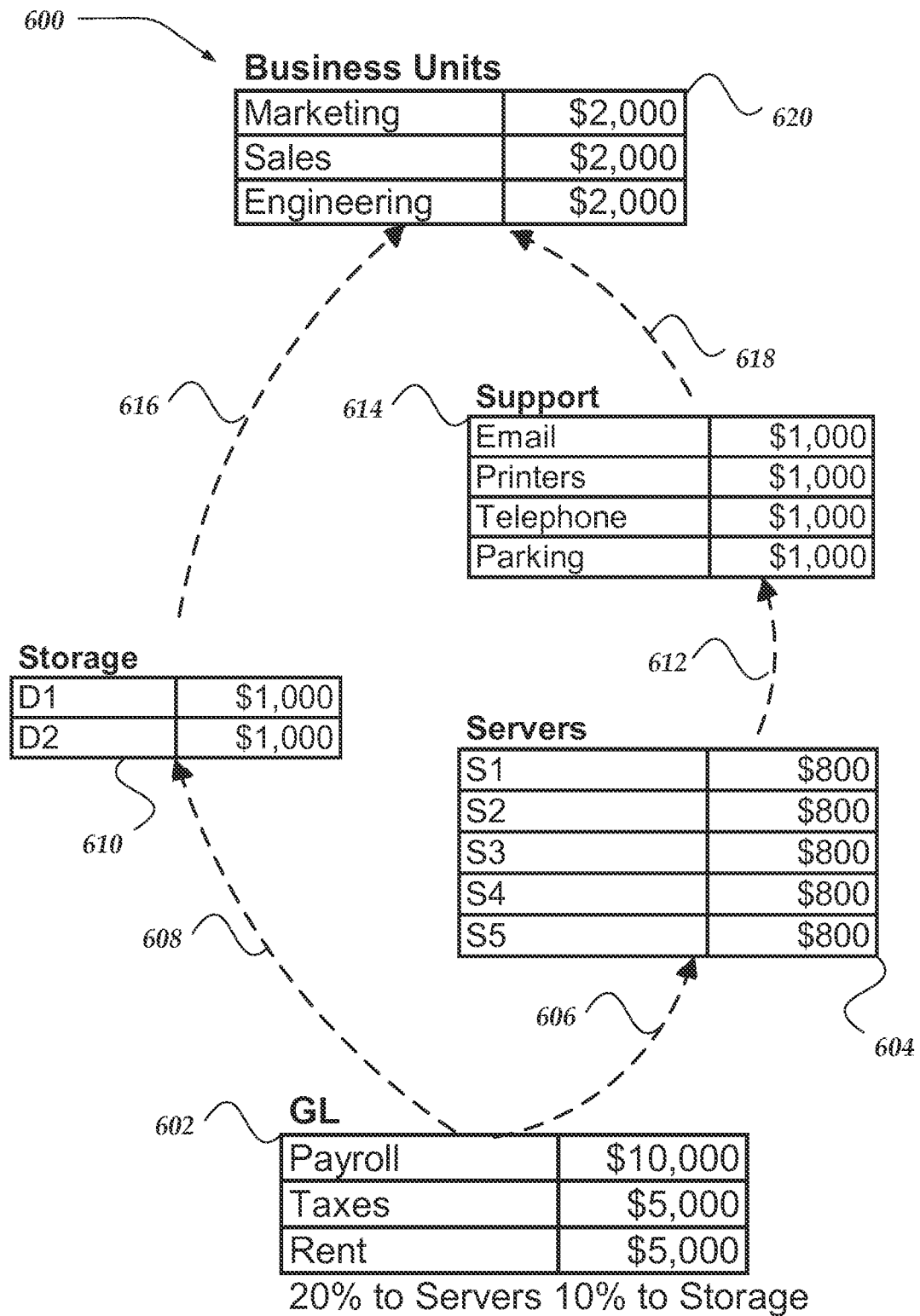


FIG. 6

7/23

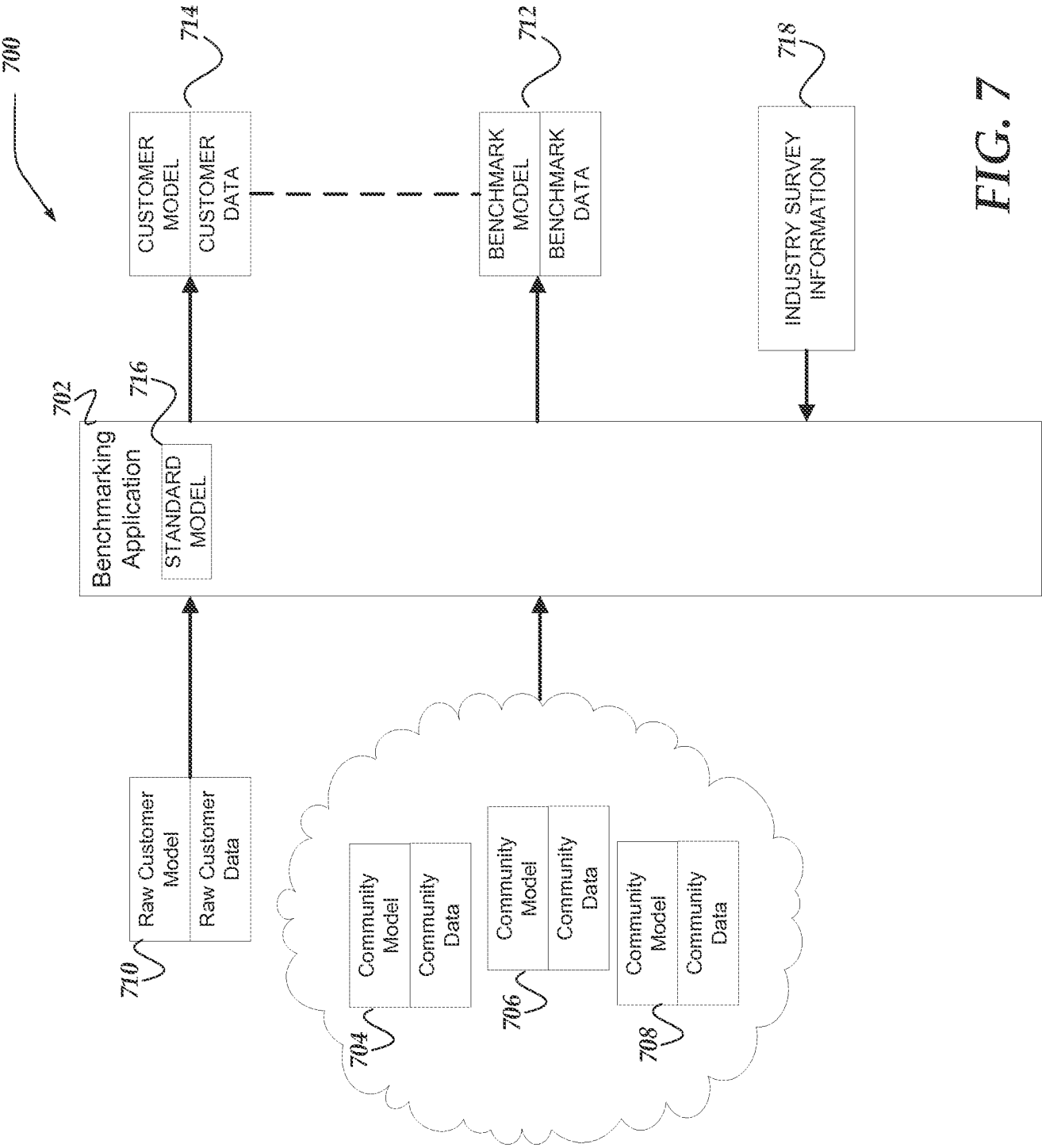


FIG. 7

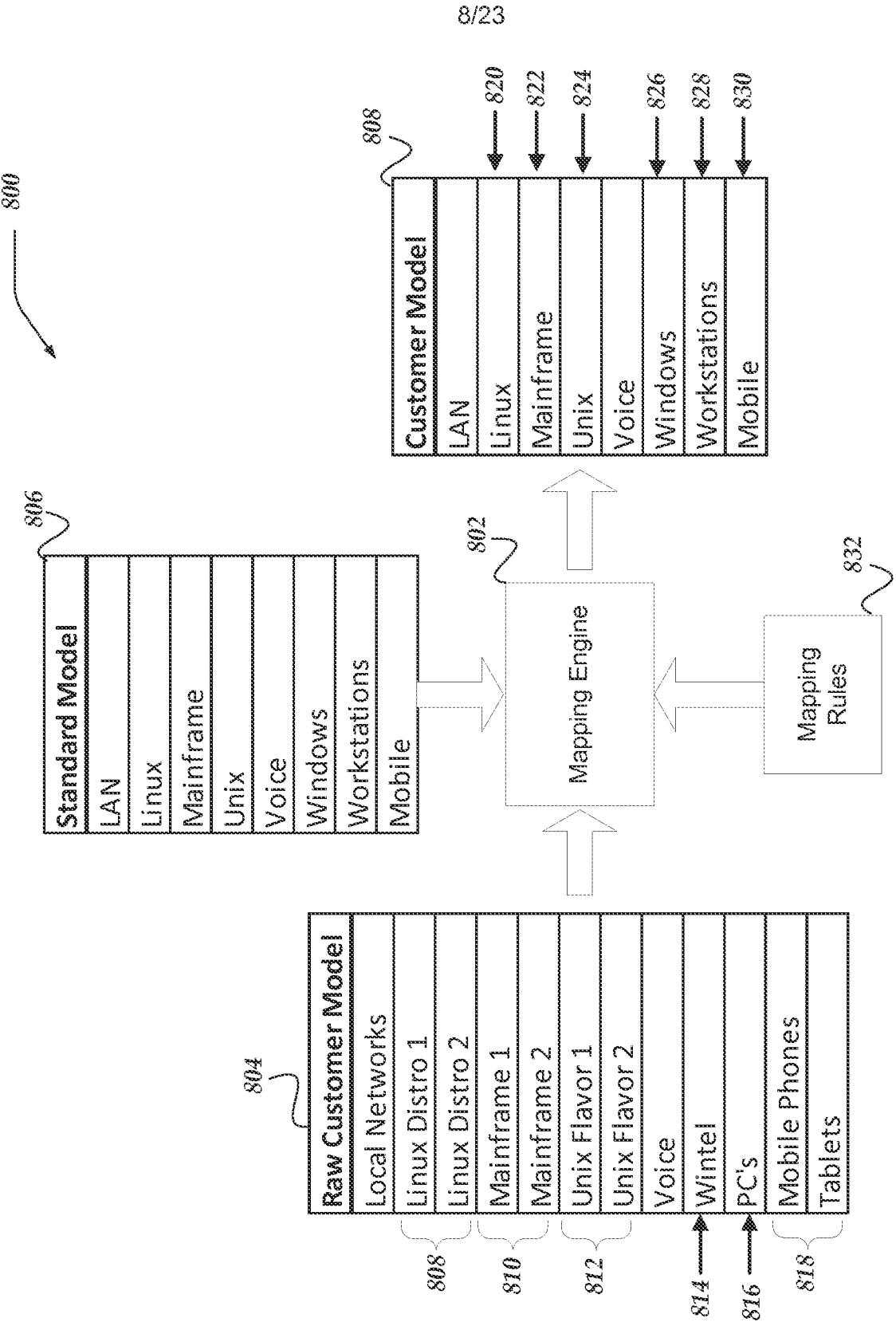


FIG. 8

9/23

900

| 902          | 904              | 906              | 908   |
|--------------|------------------|------------------|-------|
| Sub Towers   | Customer Actuals | Benchmark Values | Delta |
| LAN          | \$ 12,000.00     | \$ 10,000.00     | 20%   |
| Linux        | \$ 12,000.00     | \$ 3,000.00      | 300%  |
| Mainframe    | \$ 8,000.00      | \$ 12,000.00     | -33%  |
| Unix         | \$ 4,000.00      | \$ 2,000.00      | 100%  |
| Voice        | \$ 5,500.00      | \$ 5,000.00      | 10%   |
| Windows      | \$ 2,000.00      | \$ 10,000.00     | -80%  |
| Workstations | \$ 5,000.00      | \$ 5,000.00      | 0%    |
| iMobile      | \$ 7,000.00      | \$ 5,000.00      | 40%   |

910

912

FIG. 9

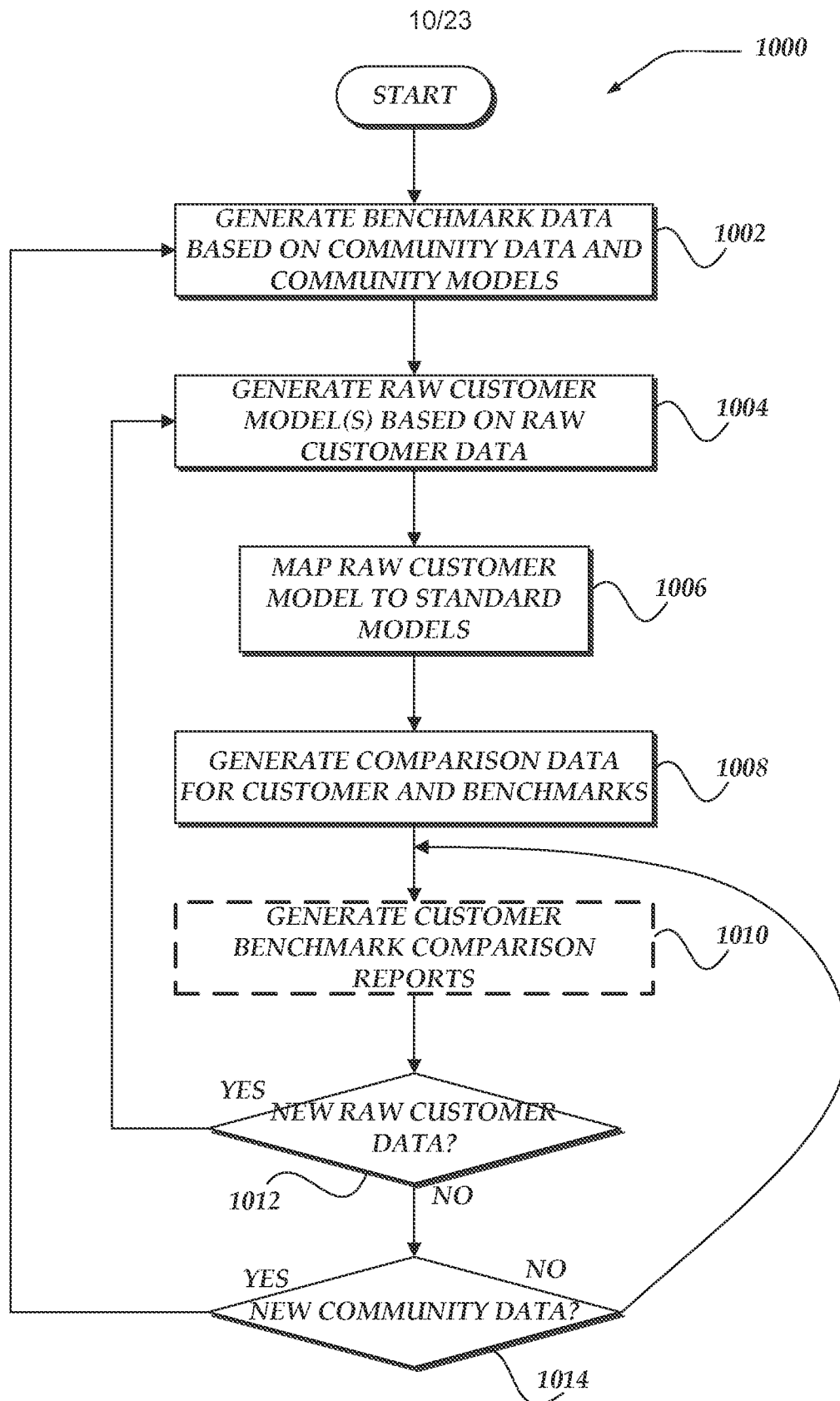
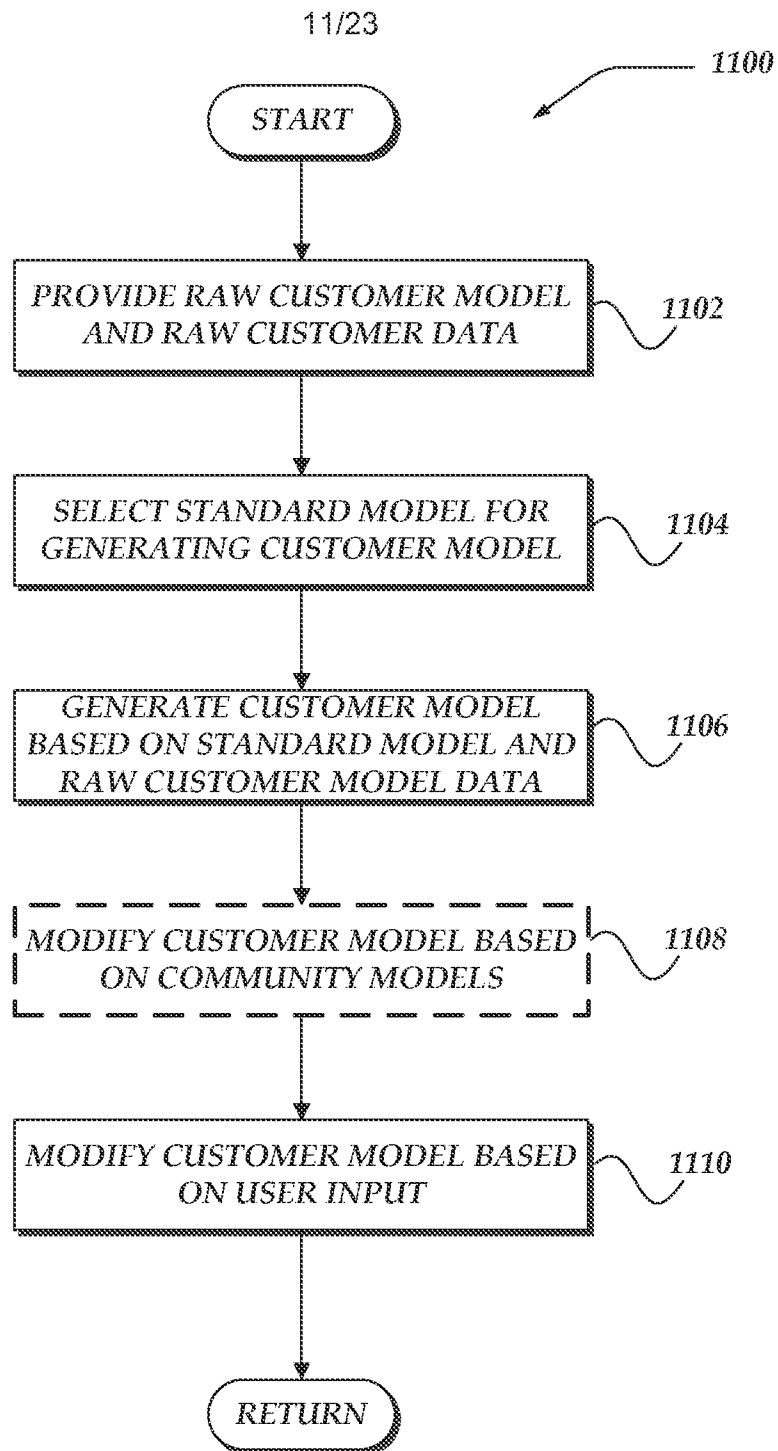
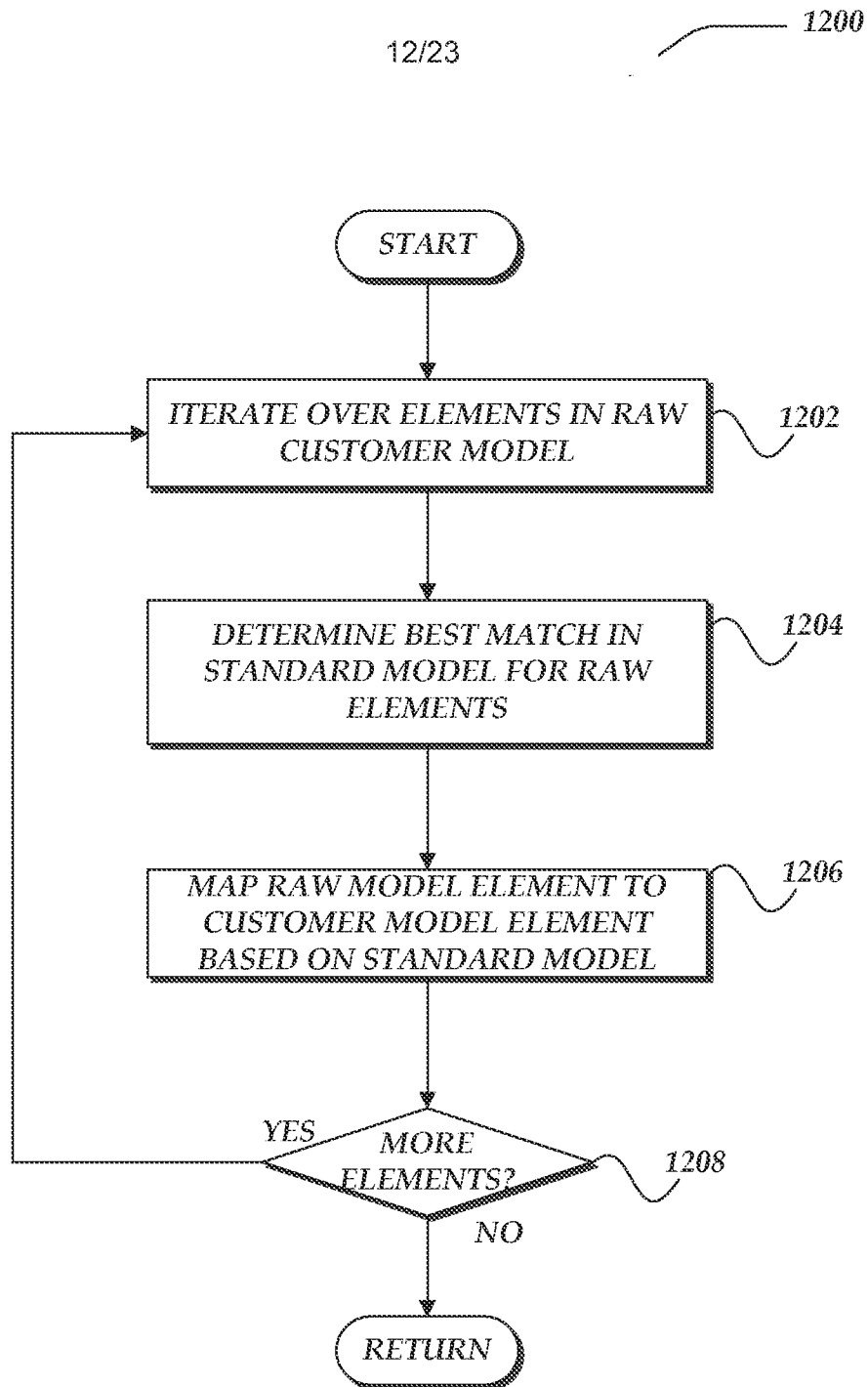
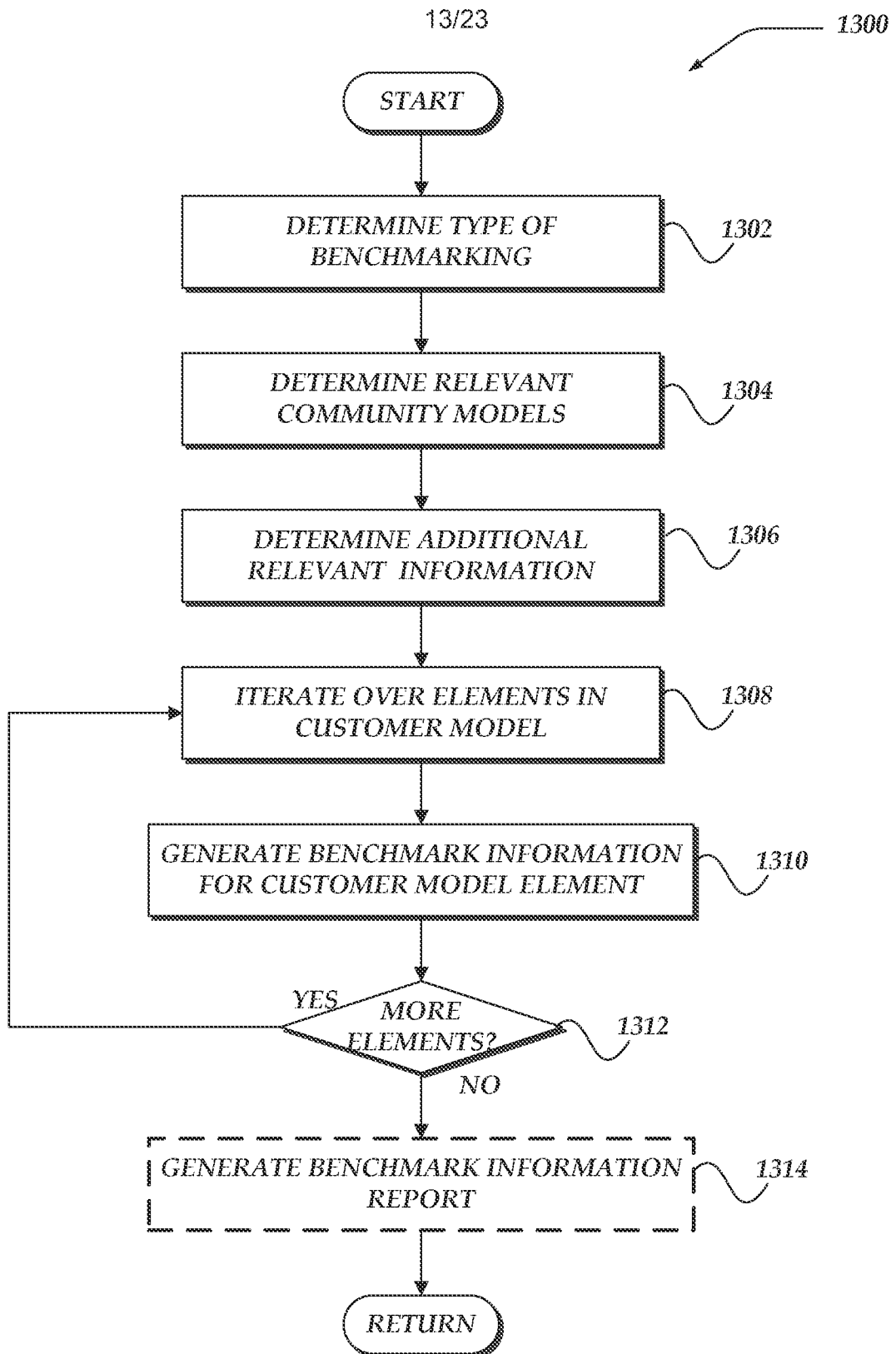


FIG. 10

**FIG. 11**

**FIG. 12**

**FIG. 13**

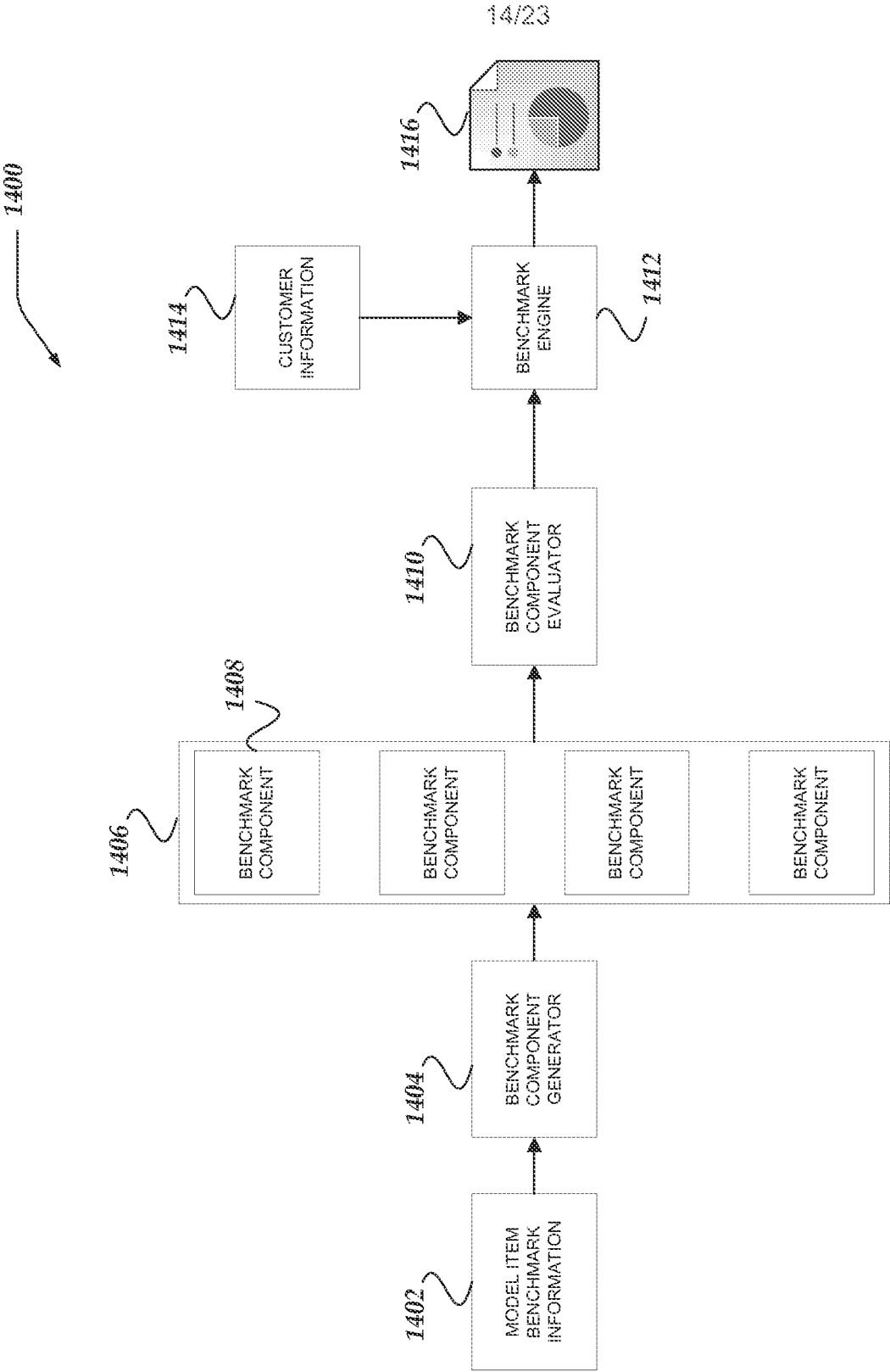
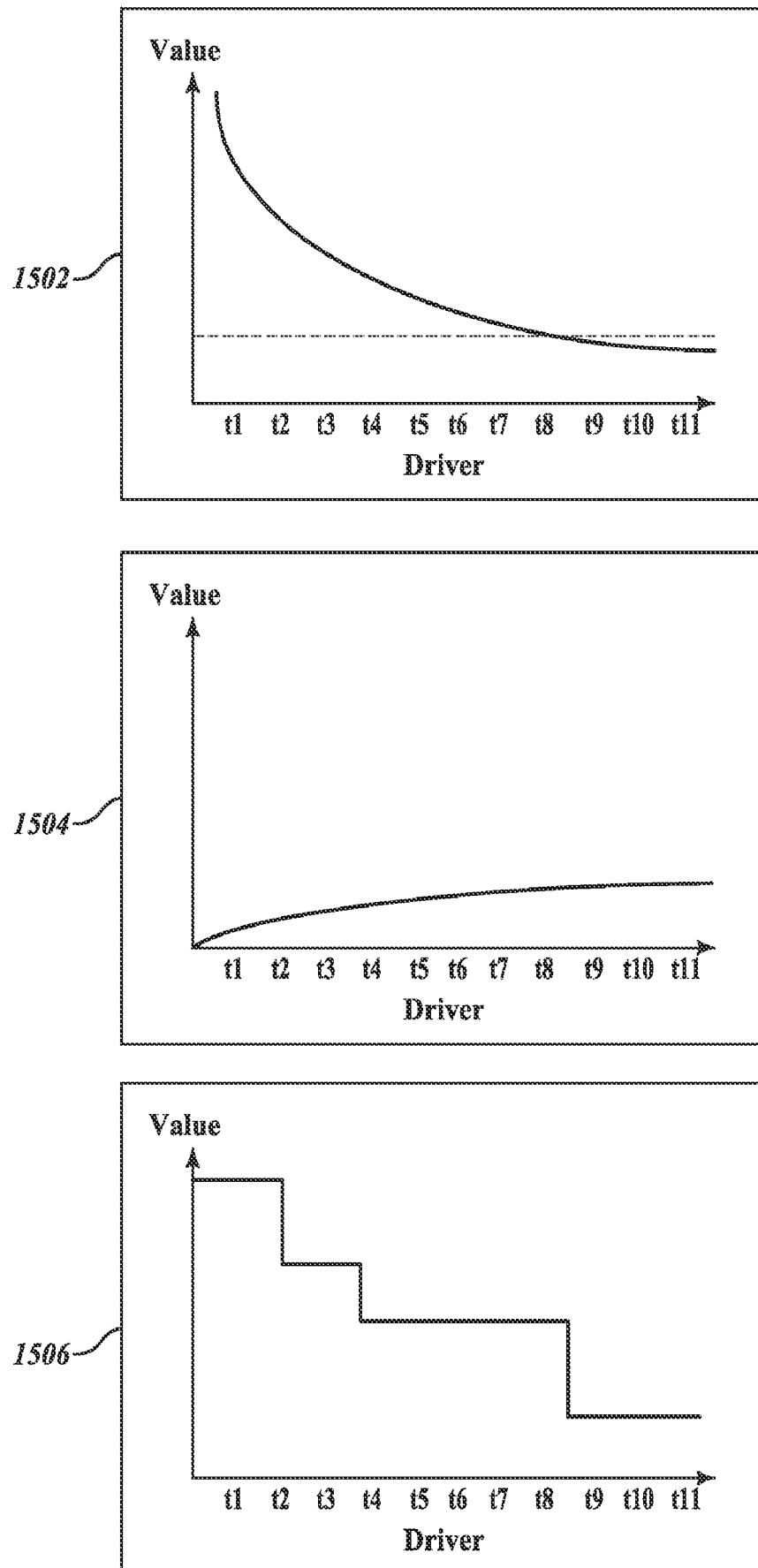


FIG. 14

15/23

**FIG. 15**

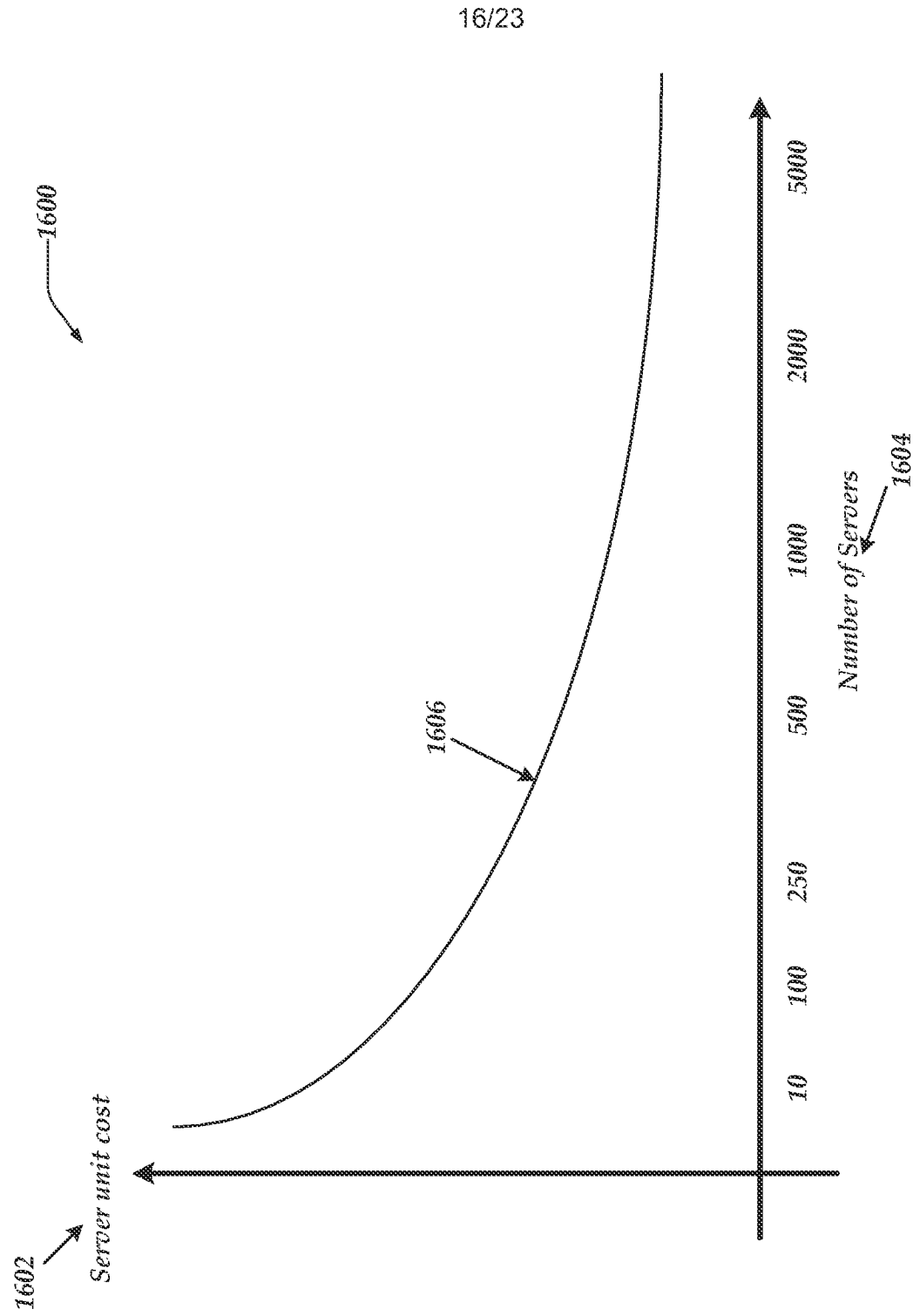
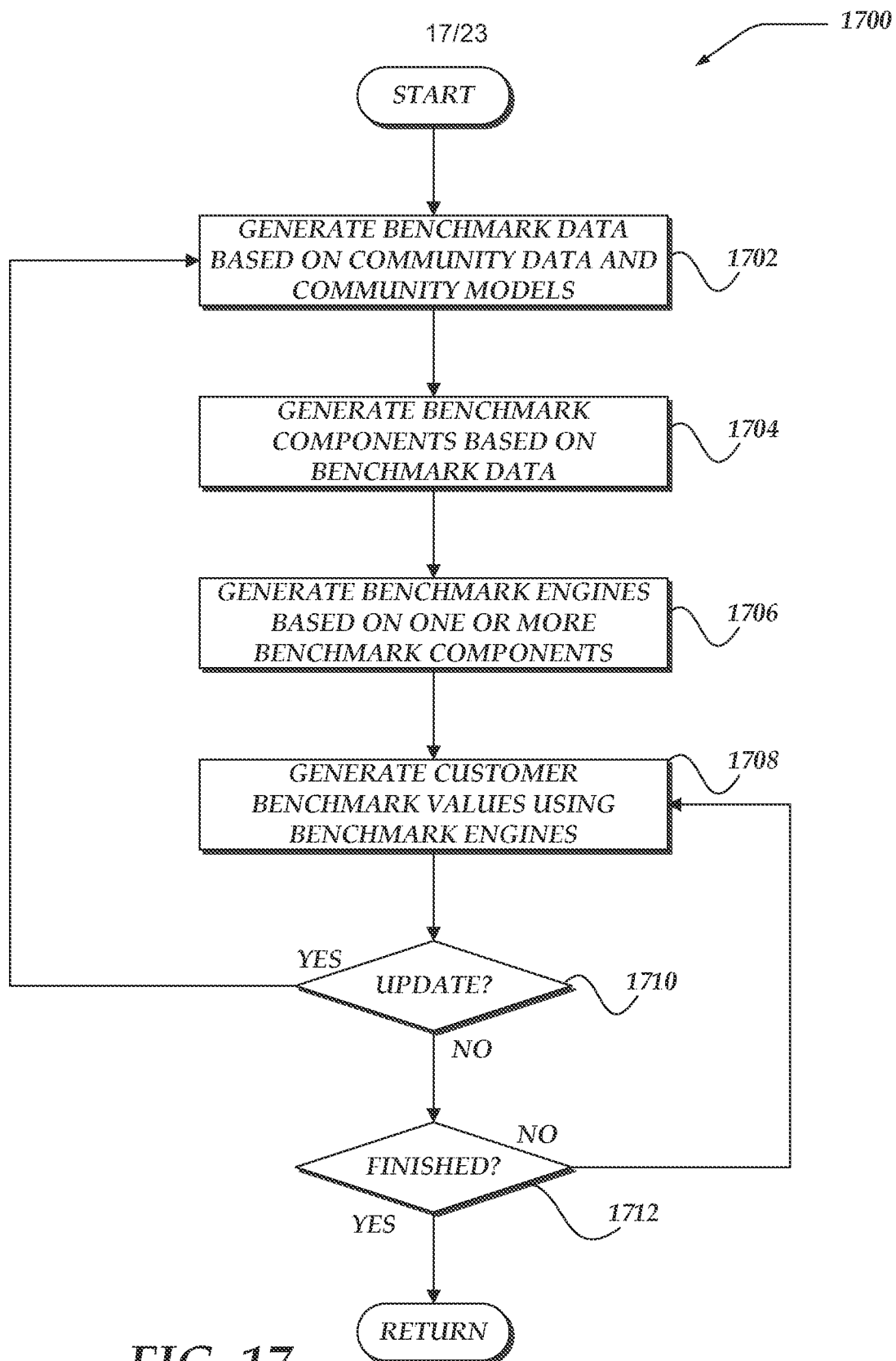
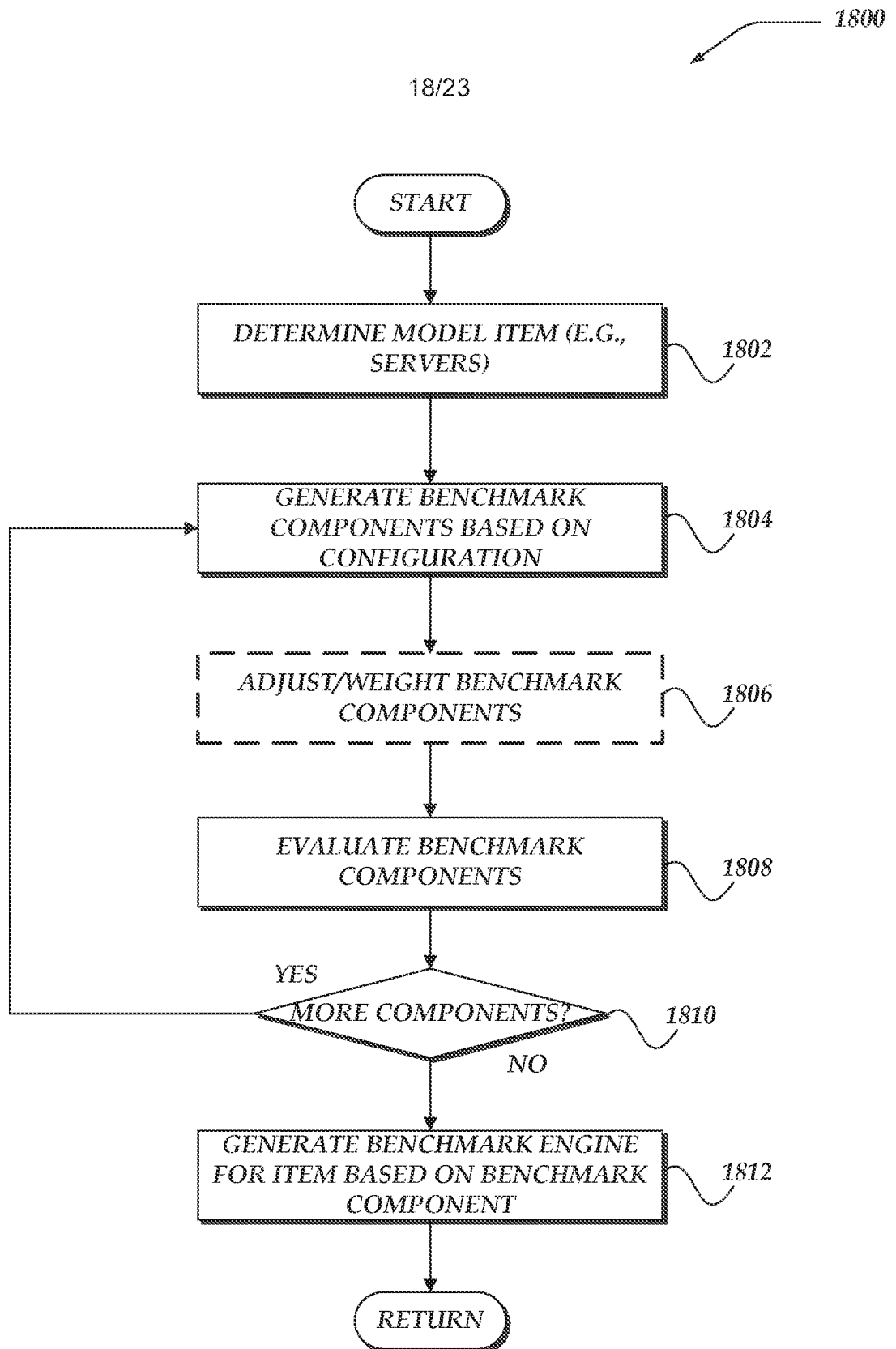


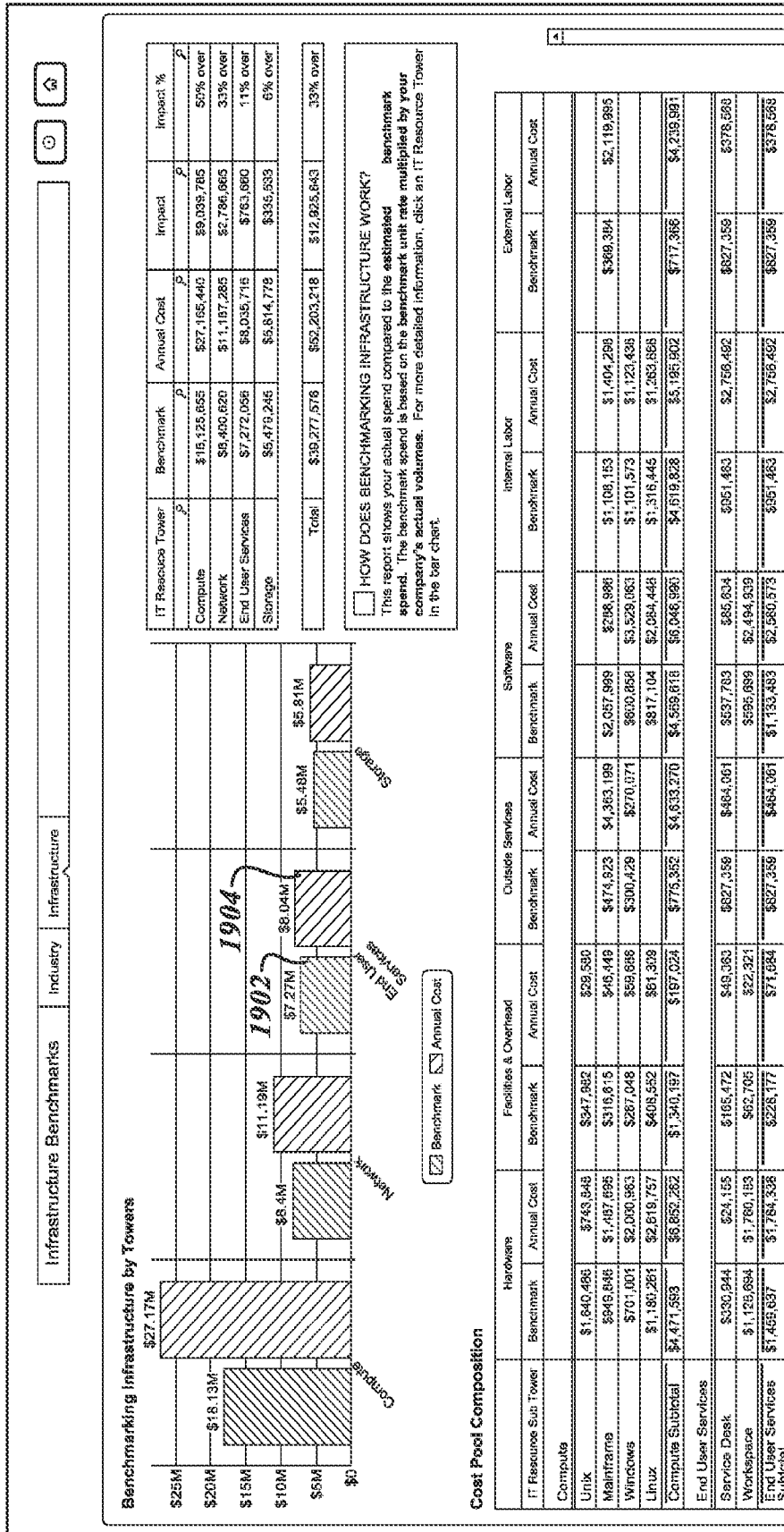
FIG. 16



**FIG. 18**

19/23

1900



**Cost Pool Composition**

| IT Resource Sub-Tower             | Hardware           |                    | Facilities & Overhead |                  | Outside Services |                    | Software           |                    | Internal Labor     |                    | External Labor   |                    |
|-----------------------------------|--------------------|--------------------|-----------------------|------------------|------------------|--------------------|--------------------|--------------------|--------------------|--------------------|------------------|--------------------|
|                                   | Benchmark          | Annual Cost        | Benchmark             | Annual Cost      | Benchmark        | Annual Cost        | Benchmark          | Annual Cost        | Benchmark          | Annual Cost        | Benchmark        | Annual Cost        |
| <b>Compute</b>                    |                    |                    |                       |                  |                  |                    |                    |                    |                    |                    |                  |                    |
| Unix                              | \$1,840,486        | \$743,848          | \$347,992             | \$29,590         |                  |                    |                    |                    |                    |                    |                  |                    |
| Mainframe                         | \$949,846          | \$1,497,895        | \$316,615             | \$46,449         | \$474,923        | \$4,363,199        | \$2,057,999        | \$1,404,296        | \$1,108,153        | \$1,404,296        | \$388,394        | \$2,119,995        |
| Windows                           | \$701,001          | \$2,000,963        | \$267,048             | \$59,888         | \$300,429        | \$270,071          | \$630,858          | \$3,529,063        | \$1,101,573        | \$1,123,436        |                  |                    |
| Linux                             | \$1,180,281        | \$2,618,757        | \$408,552             | \$61,308         |                  |                    | \$817,104          | \$2,084,448        | \$1,315,445        | \$1,263,866        |                  |                    |
| <b>Compute Subtotal</b>           | <b>\$4,471,593</b> | <b>\$6,852,262</b> | <b>\$1,340,197</b>    | <b>\$197,024</b> | <b>\$775,352</b> | <b>\$4,633,270</b> | <b>\$4,559,818</b> | <b>\$6,046,991</b> | <b>\$4,519,828</b> | <b>\$5,185,902</b> | <b>\$777,388</b> | <b>\$4,239,991</b> |
| <b>End User Services</b>          |                    |                    |                       |                  |                  |                    |                    |                    |                    |                    |                  |                    |
| Service Desk                      | \$320,944          | \$24,155           | \$165,472             | \$49,363         | \$827,356        | \$464,061          | \$537,793          | \$85,634           | \$951,483          | \$2,758,492        | \$827,356        | \$378,593          |
| Workspace                         | \$1,129,894        | \$1,780,193        | \$62,705              | \$22,321         |                  |                    | \$595,989          | \$2,494,939        |                    |                    |                  |                    |
| <b>End User Services Subtotal</b> | <b>\$1,450,837</b> | <b>\$1,784,308</b> | <b>\$228,177</b>      | <b>\$71,684</b>  | <b>\$827,356</b> | <b>\$464,061</b>   | <b>\$1,133,783</b> | <b>\$2,580,573</b> | <b>\$951,483</b>   | <b>\$2,758,492</b> | <b>\$827,356</b> | <b>\$378,593</b>   |

1906

1908

1910

1900

1902

1904

FIG. 19

20/23

2000

2006

2004

2002

Infrastructure Benchmarks

Industry

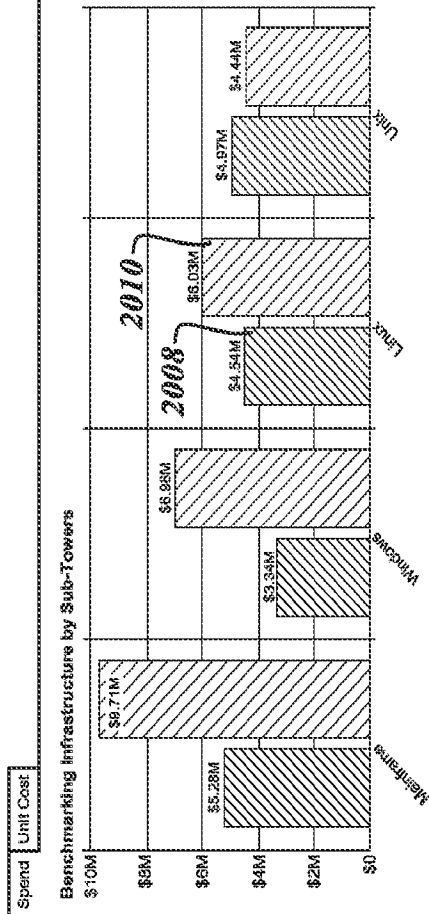
Infrastructure

Infrastructure Benchmarks

Industry

Infrastructure

Benchmarking Details - Compute Tower  
Benchmark Spend : \$18,125,655 | Annual Spend : \$27,165,440 | Impact : \$9,039,785 (50% over)



| IT Resource Sub-Tower | Benchmark   | Annual Cost | Impact      | Impact %  | Composition |
|-----------------------|-------------|-------------|-------------|-----------|-------------|
| Mainframe             | \$5,276,920 | \$8,740,623 | \$4,433,703 | 84% over  | View        |
| Windows               | \$3,338,101 | \$8,583,222 | \$3,642,121 | 108% over | View        |
| Linux                 | \$4,539,484 | \$8,029,301 | \$1,489,817 | 33% over  | View        |
| Unk                   | \$4,971,170 | \$4,442,214 | (\$528,956) | 11% under | View        |

|       |              |              |             |          |
|-------|--------------|--------------|-------------|----------|
| Total | \$18,125,655 | \$27,165,440 | \$9,039,785 | 50% over |
|-------|--------------|--------------|-------------|----------|

☐ HOW DOES THE SPEND TAB WORK?  
This report shows your actual spend compared to the estimated Aspic benchmark spend for the sub-towers in the selected tower. The benchmark spend is based on the benchmark unit rate multiplied by your company's actual volumes.

FIG.20

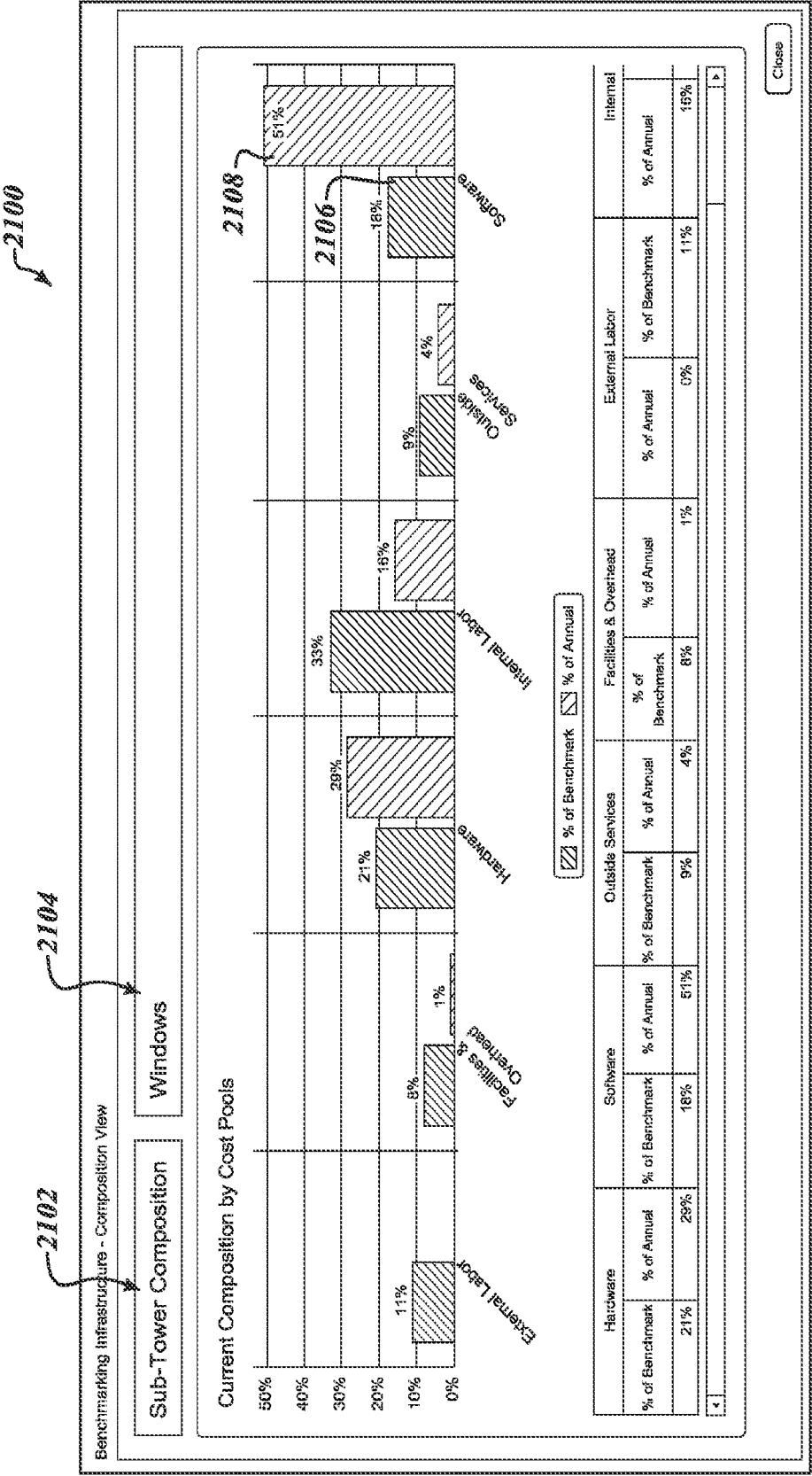


FIG. 21

|            |            |  |                           |          |                |
|------------|------------|--|---------------------------|----------|----------------|
| Sub-Towers | Cost Pools |  | Infrastructure Benchmarks | Industry | Infrastructure |
|------------|------------|--|---------------------------|----------|----------------|

**Benchmark Sub-Towers Mapped**

13 of 13

**Customer Sub-Towers Mapped**

13 of 45

**Instructions:**

- Review the *Mapped Sub-Towers* table. This shows which IT Sub-Towers will be compared to the peer benchmarks (e.g. WAN). Sub-Towers with the same name will automatically be mapped.
- Review the *Unmapped Benchmark Sub-Towers* table to identify which benchmarks are available but not yet mapped.
- Review the *Unmapped Customer Sub-Towers* table to identify which IT Sub-Tower could be mapped to the available Benchmarks Sub-Towers.
- Create one or more *Mapping Rules* to map your unmapped Sub-Towers to the available and appropriate benchmark IT Sub-Towers.

**Mapped Sub-Towers**

| Benchmark Sub-Tower | Customer Sub-Tower |
|---------------------|--------------------|
| LAN                 | LAN                |
| Linux               | Linux              |
| Mainframe           | Mainframe          |
| Service Desk        | Service Desk       |
| Tier 1              | Tier 1             |
| Tier 2              | Tier 2             |
| Tier 3              | Tier 3             |
| Tier 4              | Tier 4             |
| Unix                | Unix               |
| Voice               | Voice              |
| WAN                 | WAN                |
| Windows             | Windows            |
| PCs                 | Workpace           |

**Unmapped Benchmark Sub-Towers**

| Benchmark Tower     | Customer Sub-Tower |
|---------------------|--------------------|
| No rows to display. |                    |

**Unmapped Customer Sub-Towers**

| Customer Tower        | Customer Sub-Towers       |
|-----------------------|---------------------------|
| Application           | App Dev                   |
| Application           | App Support & Ops         |
| Communication         | Circuits                  |
| Delivery              | Client Mgmt               |
| Application           | Cloud Apps                |
| Compute               | Cloud Compute Linux       |
| Compute               | Cloud Compute Windows     |
| Security & Compliance | Cloud DR                  |
| End User Services     | Cloud Desktop             |
| Network               | Cloud Network             |
| Delivery              | Cloud Ops                 |
| Compute               | Cloud Platform            |
| Storage               | Cloud Storage             |
| Security & Compliance | Compliance                |
| Compute               | Converged Infrastructures |
| Compute               | Database                  |
| Security & Compliance | Disaster Recovery         |
| IT Mgmt               | Enterprise Architecture   |
| Data Center           | Enterprise Data Center    |

**Mapping Rules**

| Benchmark Sub-Tower | Customer Sub-Tower |
|---------------------|--------------------|
| PCs                 | Workspace          |
| Windows             | Windows            |

**FIG. 22**

2300

Infrastructure Benchmarks

Industry

Infrastructure

Sub-Towers

Cost Pools

Benchmark Cost Pools Mapped

7 of 7

Customer Cost Pools Mapped

7 of 7

Instructions:

1. Review the *Mapped Cost Pools* table. This shows which cost pools will be compared to the peer benchmark cost pool composition (e.g. Internal Labor). Cost pools with the same label as the benchmark cost pools will be automatically mapped.

2. Review the *Unmapped Benchmark Cost Pools* table to identify which benchmark cost pools are available but not yet mapped to your cost pools.

3. Review *Unmapped Customer Cost Pools* to identify your cost pools which are not and should be mapped to the available benchmark cost pools.

4. Create one or more *Mapping Rules* to map your unmapped cost pools to the benchmark cost pool composition. Mappings must be one to one.

Mapped Cost Pools

| Benchmark Cost Pool   | Customer Cost Pool    |
|-----------------------|-----------------------|
| Telecom               | Telecom               |
| Software              | Software              |
| Outside Services      | Outside Services      |
| Internal Labor        | Internal Labor        |
| Hardware              | Hardware              |
| Facilities & Overhead | Facilities & Overhead |
| External Labor        | External Labor        |

Unmapped Benchmark Cost Pools

| Benchmark Cost Pool |
|---------------------|
| No rows to display. |

Unmapped Customer Cost Pools

| Customer Cost Pool  |
|---------------------|
| No rows to display. |

Mapping Rules

| Benchmark Cost Pool | Customer Cost Pool |
|---------------------|--------------------|
| No rows to display. |                    |

FIG. 23

## INTERNATIONAL SEARCH REPORT

International application No.  
**PCT/US2015/048697****A. CLASSIFICATION OF SUBJECT MATTER****G06Q 10/06(2012.01)i, G06Q 30/06(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 10/06; G06F 15/173; G06F 17/60; G06Q 40/00; G06Q 30/06

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) &amp; Keywords: resource, allocation, model, benchmark, update

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                   | Relevant to claim No. |
|-----------|----------------------------------------------------------------------------------------------------------------------|-----------------------|
| A         | US 2005-0071285 A1 (KAROLIN LAICHER et al.) 31 March 2005<br>See abstract, claims 1-18 and figures 1,6.              | 1-24                  |
| A         | US 2012-0232947 A1 (PAUL DAMIEN MCLACHLAN et al.) 13 September 2012<br>See abstract, claims 1-10 and figures 4-8,13. | 1-24                  |
| A         | US 2007-0088641 A1 (JEFFREY A. AARON et al.) 19 April 2007<br>See abstract, claims 1-12 and figure 3.                | 1-24                  |
| A         | US 2003-0083912 A1 (ROY B. COVINGTON III et al.) 01 May 2003<br>See abstract, claims 1-2,10-12 and figure 6.         | 1-24                  |
| A         | US 2002-0194329 A1 (ERIC R. ALLING) 19 December 2002<br>See abstract, claims 1-6 and figure 4.                       | 1-24                  |



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

"&amp;" document member of the same patent family

Date of the actual completion of the international search

30 March 2016 (30.03.2016)

Date of mailing of the international search report

**31 March 2016 (31.03.2016)**

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

LEE, EUN KYU

Telephone No. +82-42-481-3580



**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

**PCT/US2015/048697**

| Patent document<br>cited in search report | Publication<br>date | Patent family<br>member(s) | Publication<br>date |
|-------------------------------------------|---------------------|----------------------------|---------------------|
| US 2005-0071285 A1                        | 31/03/2005          | US 8572003 B2              | 29/10/2013          |
| US 2012-0232947 A1                        | 13/09/2012          | EP 2684165 A2              | 15/01/2014          |
|                                           |                     | EP 2684165 A4              | 25/02/2015          |
|                                           |                     | US 2012-0233217 A1         | 13/09/2012          |
|                                           |                     | US 2012-0233547 A1         | 13/09/2012          |
|                                           |                     | WO 2012-122410 A2          | 13/09/2012          |
|                                           |                     | WO 2012-122410 A3          | 27/12/2012          |
|                                           |                     | WO 2012-122432 A2          | 13/09/2012          |
|                                           |                     | WO 2012-122432 A3          | 01/11/2012          |
| US 2007-0088641 A1                        | 19/04/2007          | US 7742961 B2              | 22/06/2010          |
| US 2003-0083912 A1                        | 01/05/2003          | None                       |                     |
| US 2002-0194329 A1                        | 19/12/2002          | None                       |                     |