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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 October 2012 (11.10.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/139098 A1

(51) International Patent Classification:
G06F 7/00 (2006.01) *G06F 17/30* (2006.01)

(21) International Application Number:
PCT/US2012/032726

(22) International Filing Date:
9 April 2012 (09.04.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/472,812 7 April 2011 (07.04.2011) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 61/472,812 (CON)
Filed on 7 April 2011 (07.04.2011)

(71) Applicant (*for all designated States except US*): **PNEURON CORP.** [US/US]; 216 Daniel Webster Highway, Nashua, New Hampshire 03060 (US).

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **BACHELOR, Douglas Wiley** [US/US]; 26 Brown Lane, Groton, Massachusetts 01450-1485 (US). **CURBELO, Raul Hugo** [US/US]; 174 Cedar Street, Sturbridge, Massachusetts 01566 (US). **ELKINS, Elizabeth Winters** [US/US]; 1255 North Shore Drive, Roswell, Georgia 30076 (US). **MCGRATH, Christie Patrick** [US/US]; 121 Cherrywood Drive, Nashua, New Hampshire 03062 (US). **MOSS, Simon Byford** [GB/US]; 210 Bible Street, Greenwich, Connecticut 06807 (US).

(74) Agents: **BOURQUE AND ASSOCIATES, PA** et al.; 835 Hanover Street, Suite 301, Manchester, New Hampshire 03104 (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, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

[Continued on next page]

(54) Title: LEGACY APPLICATION MIGRATION TO REAL TIME, PARALLEL PERFORMANCE CLOUD

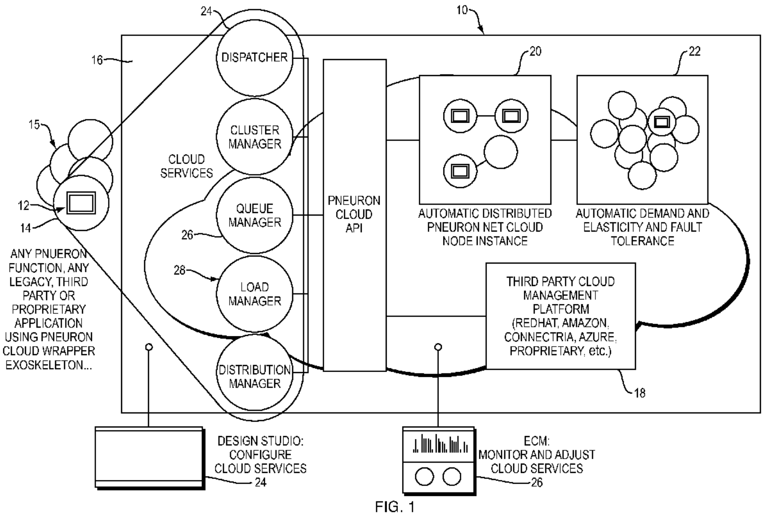
24
10
16
20
22
DISPATCHER
CLUSTER MANAGER
CLOUD SERVICES
15
12
14
QUEUE MANAGER
26
PNEURON CLOUD API
AUTOMATIC DISTRIBUTED PNEURON NET CLOUD NODE INSTANCE
AUTOMATIC DEMAND AND ELASTICITY AND FAULT TOLERANCE
ANY PNUERON FUNCTION, ANY LEGACY, THIRD PARTY OR PROPRIETARY APPLICATION USING PNEURON CLOUD WRAPPER EXOSKELETON...
28
LOAD MANAGER
THIRD PARTY CLOUD MANAGEMENT PLATFORM (REDHAT, AMAZON, CONNECTRIA, AZURE, PROPRIETARY, etc.)
DISTRIBUTION MANAGER
18
DESIGN STUDIO: CONFIGURE CLOUD SERVICES
24
ECM: MONITOR AND ADJUST CLOUD SERVICES
26

FIG. 1

(57) **Abstract**: A system and method 10 that provides legacy software applications 12 to be incorporated into a massively parallel and distribution processing model in the Cloud, with a high performance parallel and distributed computing (cloud) "wrapper" 14 around legacy and current systems to enable, without legacy product code change or invasive addition/editing, the legacy product to access and fully utilize the power and ability of distributed computing within the cloud through a Pneuron "cortex" platform virtual server 10. The system and method also provides the ability to distribute multiple, concurrent instances of the legacy applications, dynamically manage the load volumes, and automatically create and remove new virtual machines based upon demand requirements.

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

LEGACY APPLICATION MIGRATION TO REAL TIME, PARALLEL PERFORMANCE CLOUD

TECHNICAL FIELD

[0001] The present invention relates to cloud computing, parallel processing, distributed computing, fault tolerance, and load balancing and more particularly, relates to a system and method for providing legacy software applications to be incorporated into a massively parallel and distribution processing model in the Cloud

BACKGROUND

[0002] Cloud computing is considered one of the top transformational IT changes and trends in 2010 and onwards. Cloud computing provides a scalable, dynamic, and distributed infrastructure for enabling applications to dynamically obtain and utilize computing resources on demand.

[0003] Most businesses have existing computer software applications that are not engineered or architected for cloud computing. Many existing applications cannot perform distributed and parallel processing and are unable to be deployed on an elastic cloud computing environment without significant changes to the existing applications' source code and application architecture. The challenge is applying a cost effective and simple migration and transformation process for legacy applications and products so that they can be incorporated into the cloud computing environment.

[0004] A legacy system or application program is a previously deployed third party or internally developed customer application that continues to be used, typically because it still functions for the users' needs or is too expensive to replace, even though newer technology or more efficient methods of performing a task are now available. Legacy applications include Java command line programs, Java thin client and server based applications, Microsoft thin client and server-based applications, Client/Server applications, Client workstation applications, third party independent software vendor applications, and proprietary client applications running on proprietary architectures and operating systems. Legacy application programs are typically serially programmed, linear in performance, residing in only one data center and with a limited number of users constrained by geographic or business location.

[0005] To implement an existing legacy application in the cloud computing environment and enable the application to be distributed, parallel, and demand-elastic is a very expensive and time-consuming activity. The existing application architecture and source code needs to be re-factored and significantly rewritten, tested extensively, and re-integrated with its existing applications.

[0006] The cost and time to implement such a legacy application re-write can exceed the original cost and development time. Given these impediments, businesses are unable to adapt their existing applications to the cloud computing environment. The new Cloud paradigm offers huge business and technical value including parallel processing, distributed, use anywhere, high performance, high resiliency, high accessibility and high availability.

[0007] This significant divergence in technology deployment and accessibility creates a significant challenge: the cost effective and manageable ability to migrate applications, best processes and institutionalized intellectual property from serial, linear, data center or application confined applications to a new technology paradigm of agile, competitive, massively performed, use anywhere and broadly distributed/accessible functions. It is this challenge that the invention specifically addresses.

[0008] Accordingly, the objective of the present invention is to provide a non-intrusive and assimilation model which enables these legacy applications to be "transformed" from standalone, serial-based computing to highly distributed, parallel processing and cooperating cloud application services. The innovation combines the cloud-centric, distributed, and parallel processing Pneuron platform infrastructure with a suite of Pneuron legacy wrapper libraries to assimilate or "wrap" and encapsulate legacy applications and enable them to operate within the Pneuron cloud computing platform.

[0009] The present invention incorporates technology and process innovation, using the assignee's distributed "Pneuron platform" computing design and high performance, highly resilient virtual server for the migration of legacy applications to cloud that is simple, quick and a fraction of the cost of traditional migration and porting solutions and referred to herein as the "Pneuron platfom" or "Pneuron Platform and technology" or "Pneuron technology" or simply "Pneuron".

SUMMARY

[0010] .The present invention relates to cloud computing, parallel processing, distributed computing, fault tolerance, and load balancing and more particularly, relates to a system and method for providing legacy software applications to be incorporated into a massively parallel and distribution processing model in the Cloud having the ability to provide simple, real time, high performance parallel and distributed computing (cloud) "wrapper" around legacy and current systems to enable, without legacy product code change or invasive addition/editing, the legacy product to access and fully utilize the power and ability of distributed computing within the cloud through the Pneuron "cortex" platform virtual server.

[0011] The present invention also provides a system and method to distribute multiple, concurrent instances of the legacy applications, dynamically manage the load volumes, and automatically create and remove new virtual machines based upon demand requirements.

[0012] The system and method of the present invention facilitates inter-processing connections between distributed applications, enabling the passing of information and execution of processing across participating application nodes. Also disclosed is and innovative process for incorporating these applications into its distribution framework and enables comprehensive processing from work initiation through completion.

[0013] The present invention further features a system and method for the alignment of a business process for the use of the Pneuron platform and technology to enable refactoring and migration of legacy applications and traditional products systems into a high availability, fault tolerant, load balancing, parallel and distributed processing within a cloud virtualization computing environment. Adding to the original distributed data and analysis, this addition adds the ability to incorporate and distribute entire applications using Pneuron technology and enable existing applications to realize quickly and cost effectively the value of modern distributed cloud computing.

[0014] One key to the design of the present invention is the ability to rapidly encapsulate and incorporate any and all non-cloud computing applications into a highly elastic virtual infrastructure within the Cloud, without significantly modifying the application source or applying invasive changes to the application.

[0015] The unique process, combined with the integration to the Pneuron platform, enable speed in deployment and rapid ability to apply a parallel, distributed cloud processing model to a current "legacy" application with no reprogramming requirements, combined with industrial strength security, agility, resiliency and consumerability that forms one of the basis of this invention.

DESCRIPTION OF THE DRAWINGS

[0016] These and other features and advantages of the present invention will be better understood by reading the following detailed description, taken together with the drawings wherein:

[0017] Fig. 1 is a schematic diagram of a neuron cloud environment in which a pneuron wrapped legacy application interacts in accordance with the present invention;

[0018] Fig. 2 is a flowchart illustrating the legacy application wrapper invocation model in accordance with the teachings of the present invention;

[0019] Fig. 3 is a diagram and flowchart illustrating the runtime processing model for a legacy application wrapped with a pneuron operable on the pneuron based cloud system in accordance with the teachings of the present invention; and

[0020] Fig. 4 is a schematic diagram and flowchart of a cloud integration model showing the interaction between master pneurons, slave pneurons and worker pneurons in accordance with the teachings of the present invention.

DESCRIPTION OF THE INVENTION

[0021] The attached drawings and description (beginning with Fig. 1) highlight the Pneuron server (“Cortex") environment 10 in accordance with the present invention and its ability quickly, cost effectively and seamlessly “wrap" existing products 12 with a high performance API 14 and connect seamlessly to the Pneuron Cloud Virtual Server (Cloud Cortex) 16. The Pneuron server 16 has an interface to commercial cloud management systems 18. Different cloud management systems and environments can be supported by extending the cloud management adapter library and not affecting the overall Pneuron platform. In this manner, cloud services can be monitored and adjusted 26 while also enabling could services to be configured 24.

[0022] The external interface (wrapper API 14) enables an existing software product 12 to fully leverage massively parallel, highly distributed, horizontally and vertically resilient and available scale and so increase in performance through Pneuron parallel processing, automatically clone multiple instances 20 of the same application to ensure volume and load is managed in parallel (rather than serially in current solutions) and enable access and instantiation of the application anywhere within the cloud while maintaining automatic demand and elasticity and fault tolerance 22.

[0023] The present invention features an innovative process and technology within its Pneuron platform service that enables legacy/third party non-cloud designed software applications to be incorporated into a massively parallel and distribution processing model in the Cloud. The process is made up of a number of simple steps using the Pneuron technology wherein:

[0024] A. Existing legacy applications are "wrapped" with a Pneuron service Interface or "Wrapper"/"Exoskeleton". These "wrappers" are incorporated into a Pneuron wrapper library and preconfigured and enable "off the shelf" available interface instantiations specifically mapped to the target application's API. The "wrapper" is best considered an exoskeleton connection service which includes an encapsulation of an API, all base Pneuron functions and characteristics, security logic specific to the technology and services enabling the connection to the hyper fast, resilient, clustered Pneuron "cortex" Platform residing in the cloud. Wrappers are off the shelf and aligned to the specific type of technology including:

[0025] Java command-level programs;

[0026] Java Thin client server based applications;

[0027] Windows Thin client server based applications;

[0028] Fat Client applications;

[0029] Third Party applications; and

[0030] Mainframe applications.

[0031] Each "wrapper", once deployed, allows applications to be cloud enabled including parallel processing, on-demand vertical and horizontal scale, auto-cloning to enable load sharing and deep resiliency, high availability and clustering in either private or public cloud.

[0032] The Pneuron solution is agnostic to the interface type or application requirements - Database, a Service, an abstract, table interface, HTTP sniffer or any number of methods to ensure maximum performance and connectivity.

[0033] These wrappers, in simple terms, enable different applications to interact transparently with the Pneuron platform.

[0034] An example of a wrapper invocation model is provided as Fig. 2. The execution plan concept is as follows:

[0035] invoke(method1, paramlist)

[0036] invoke(method2, paramlist)

[0037] …

[0038] Invoke(method3, paramlist)

[0039] if required, the present invention can store partial results from previous executions and use those as parameters depending on the complexity of the operation at hand. The invoke method shown in figure 3 is based on the reflection process. In this way, the client can provide jar's or provide from the beginning a collection of Java source files that will be incorporated in the project from an early stage, or simply have them added at runtime as a jar import. In this way, the client has the possibility to use different calls from different class instances (actually object instances for those classes) and organize those in a way (instruction code block) that will allow the client to organize the logic and create the proper integration of the code.

[0040] A sample call (concept) is shown below:

[0041] Aux inst1val=objConf1instance.invoke("methodconfigname1", paraminputList)

[0042] Aux inst2val=objConf1instance.invoke("methodconfigname1", paraminputList.Inst1val)

[0043] ….

[0044] Rez instWval=objconfxinstance.invoke("methodw",parameters)

[0045] ….

[0046] Rez instNval=objconfxinstance.invoke(“methodn”,parameters)

[0047] Result=OriginalXmlRequest + InstWval + InstNval (tags)

[0048] Once the applications are “wrapped", the applications are represented as Pneurons 15, Fig. 1, within the Pneuron Platform. The applications can then be incorporated into Pneuron networks and linked to the different Pneuron applications, including the Design Studio 24, Enterprise Control Manager 26, and the Admin application 18. The wrapped legacy Pneurons 15 now have configuration plans established. During the runtime processing, the Pneuron “cortex" platform orchestrates the remote instances of the configured Pneuron networks and enables the target legacy application to initiate multi-threaded requests, transmit and fully utilize a high availability, elastic and resilient architecture.

[0049] This “cortex" platform is a high performance, cloud enabled virtual server that enables the wrapped legacy application 15 to dynamically create and remove new cloud nodes and automatically deploy new application images with remote instances of the legacy application operating within the Pneuron platform. Specific functions applied include: auto load share; distribution and accessibility across multiple data centers to utilize capacity anywhere within the cloud; auto-clone as and when necessary for automated capacity and performance balancing; scale vertically and horizontally without manual initiation; auto-restart in different location if the application encounters critical errors or failure to operated; and seamlessly integrates with other applications within the cloud as if the application has been fully and completely reconfigured for distributed, virtualized cloud operation.

[0050] The API Wrapper Pneuron 14 provides a common interface and facility for receiving and processing external application requests. The API Pneuron marshals these requests to the Dispatcher Pneuron 24 for evaluation and processing.

[0051] The Dispatcher Pneuron 24 transmits the durable messaging requests to the Intelligent Load Balance 28 and Queue Management 26 Pneurons, which are monitoring the Pneuron remote worker instances and work queues and assigning work to the legacy applications. Based on load evaluation, the Intelligent Load Balance and Queue Management Pneurons instantiates new cloud nodes based on demand and automatically propagate another remote instance and distributes work.

[0052] The legacy application wrapped Pneuron 15 performs its work and then interacts with external applications automatically. The API Wrapper Pneuron interacts with Master Pneuron components, which manage requests, keep counts of work in process, perform load balancing, marshal work to distributed wrapped legacy application instances, instantiate new instances of the legacy applications and remove instancesased on load and threshold levels.

[0053] External transaction requests 102, Fig. 3, occur through HTTP requests 104. External transaction requests will initiate an API request to the Web Service Client (JAR) 106. The Web Service Client 106 takes the WSDL and turns it into a Java call 108. The Java call request message 108 will be stored in the database. All units of work are persisted in a Pneuron persisted queue 109 for resilience, elasticity and high availability.

[0054] The Web Service Client 106 sends a JMS broadcast 110 to subscribed Master and Slave Dispatchers 112. There can be any number of Masters and Slaves and generally the number is a result of the amount of volume at any given time. This enables a dynamic creation and destruction of Pneurons which in turn allows real time volume spikes to be managed simply, allows for maximum use of available system resources no matter where within the cloud they reside and enables real time application management. A more detailed view of the master, slave and worker pneuron relationship is shown in Fig. 4.

[0055] The Master API obtains an instruction from the wrapped application (a message, a volume spike, a system crash, request to access from a location or any number of other instructions) and forwards it to the Dispatcher Pneuron. The Dispatcher Pneuron communicates with the Load and Queue Manager. If the Dispatcher is no longer the primary master - for example it goes down or the Hardware or data center supporting it is no longer accessible, the Slave will take over, applying the required units of work or activity spooling up more processing power, moving a message, cloning the application, providing secure access to the application to the requesting remote user.

[0056] The Load and Queue Manager keeps a running count of all units of work - or instructions and activities - and has to figure out which Pneuron Server 114 to hand it to for processing. Based on round robin and thresholds, it will hand the message to a Worker Instance. If all workers are busy or otherwise not available (for any reason), the Load and Queue Manager instantiates a new remote instance and server in real time. Based on thresholds, low work volume would remove remote instances and servers. The real time creation and destruction of instances, servers and nodes is a direct calibration to the volume of instructions and activities within the platform.

[0057] Inter-process communications occurs between the Master Dispatcher and the Slave Dispatcher to ensure the viability of the Dispatcher. If the communications are disrupted, the Slave Dispatcher will become the new Master dispatcher and a second take Slave over as Dispatcher the Master will be instantiated. This auto-hierarchy enables deep resiliency - if the Master fails (a server failure or data center crash)and the Master can no longer communicate and control the network, its queue, functions and all activities are instantly passed to a Slave which automatically becomes a Master.

[0058] The only time the Slave Dispatcher will get involved is switching from the master and it will use the persisted queue checkpoint 109 to start processing. When the slave becomes the master, it just re-queues everything.

[0059] The unit of work is passed from the Master Pneuron Server to the Remote Pneuron Server and the legacy wrapped Pneuron. The legacy wrapped Pneuron will perform the work. Multiple requests in flight are units of work. You can have applications or activities and have multiple Pneuron Servers. within a Pneuron Servers.

[0060] The Pneuron network will do the work directly. The load and capacity increasing and decreasing the Remote Pneuron Servers will similarly destroy and instantiate new Remote Pneuron Servers and un-persist the message from the queue (apply a dual message). The complete message Pneuron decrements the in-flight count to -1 (the in-memory queue).

[0061] Wrapped legacy applications 15 communicate and interface through Pneuron messaging protocols and initiates external transaction updates.

[0062] The benefits of the present invention include the ease of wrapping and incorporating existing applications into Cloud and distributed parallel processing infrastructure; rapid migration and transition to cloud infrastructures; automatic load and distribution of processing with platform-enabled instantiation of work queues, applications, and parallel processing on demand; and the ability to simultaneously process multiple concurrent applications sessions, reducing bottlenecks and accelerating complex, high resource processing into significantly shorter turnaround (processing jobs that take days can be reduced to hours and less).

[0063]

Modifications and substitutions by one of ordinary skill in the art are considered to be within the scope of the present invention, which is not to be limited except by the allowed claims and their legal equivalents.

CLAIMS

The invention claimed is:

1. An automated and computerized method of providing a legacy software application in a cloud computing environment, said method comprising the acts of:

receiving by an automated computer system a legacy software application to be deployed for use in a cloud computing environment;

encapsulating said received legacy software application with an application programming interface (API) wrapper by said automated computer system;

deploying said encapsulated legacy software application in a cloud computing environment; and

automatically cloning multiple instances of said deployed, encapsulated legacy software application by said automated computer system in response to multiple user requests to utilize said deployed to, encapsulated legacy software application.

2. A computerized system for providing a legacy software application operable and a cloud computing environment, said system comprising:

an automated computer system, for receiving a legacy software application to be deployed for use in a cloud computing environment;

one or more application programming interface wrappers, wherein said automated computer system is responsive to said one or more application programming interface wrappers and to said received legacy software application, for encapsulating said received legacy software application with said application programming interface (API) wrapper by said automated computer system;

responsive to encapsulating said received legacy software application with said application programming Interphase wrapper, deploying said encapsulated legacy software application in a cloud computing environment; and

responsive to multiple user requests to utilize said deployed, encapsulated legacy software application, said automated computer system automatically cloning multiple instances of said deployed, encapsulated legacy software application.

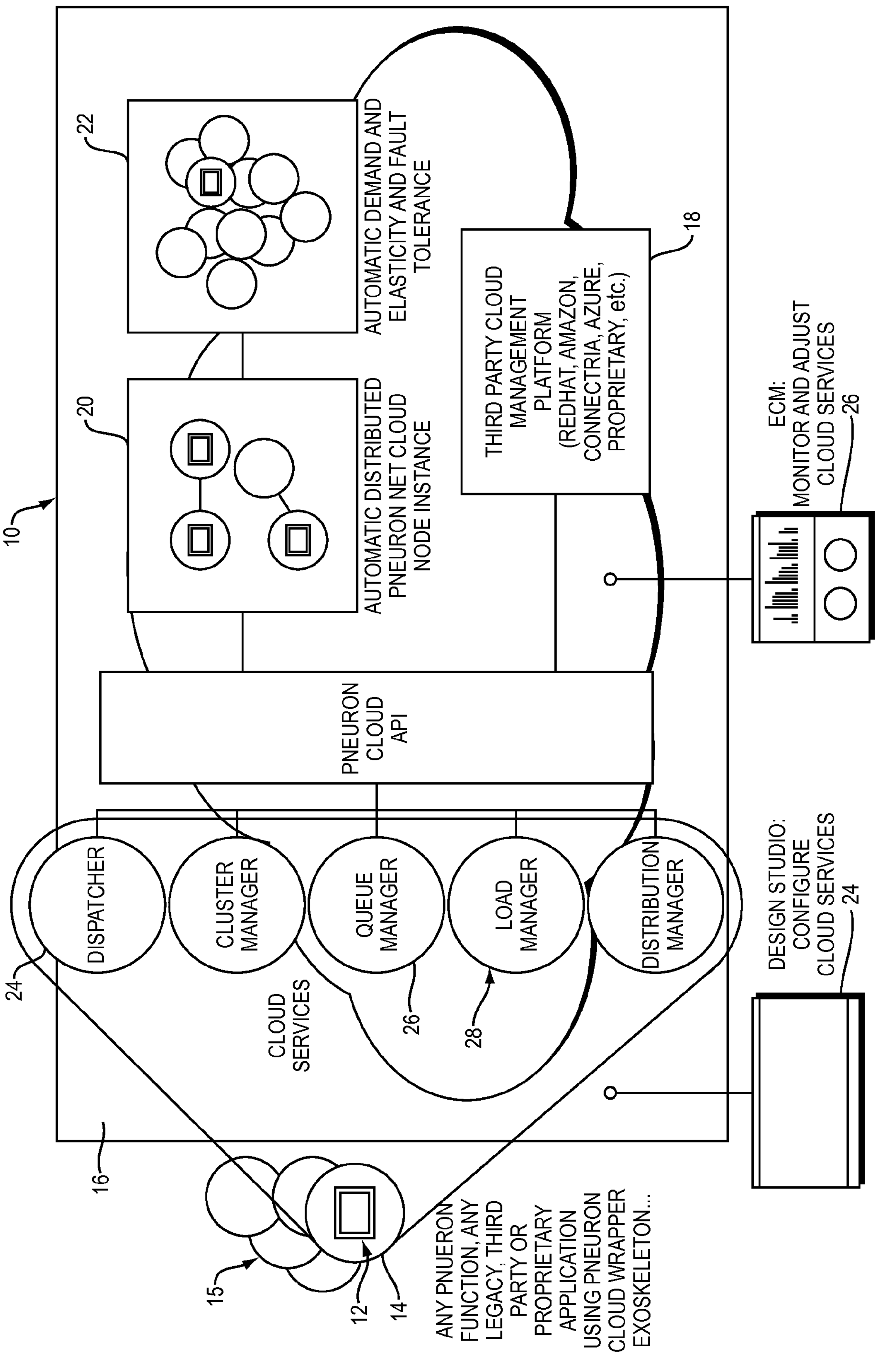
10
22
20
18
AUTOMATIC DEMAND AND ELASTICITY AND FAULT TOLERANCE
AUTOMATIC DISTRIBUTED PNEURON NET CLOUD NODE INSTANCE
THIRD PARTY CLOUD MANAGEMENT PLATFORM (REDHAT, AMAZON, CONNECTRIA, AZURE, PROPRIETARY, etc.)
ECM: MONITOR AND ADJUST CLOUD SERVICES
26
PNEURON CLOUD API
DISPATCHER
CLUSTER MANAGER
QUEUE MANAGER
LOAD MANAGER
DISTRIBUTION MANAGER
24
CLOUD SERVICES
26
28
DESIGN STUDIO: CONFIGURE CLOUD SERVICES
24
16
15
12
14
ANY PNUERON FUNCTION, ANY LEGACY, THIRD PARTY OR PROPRIETARY APPLICATION USING PNEURON CLOUD WRAPPER EXOSKELETON...

FIG. 1

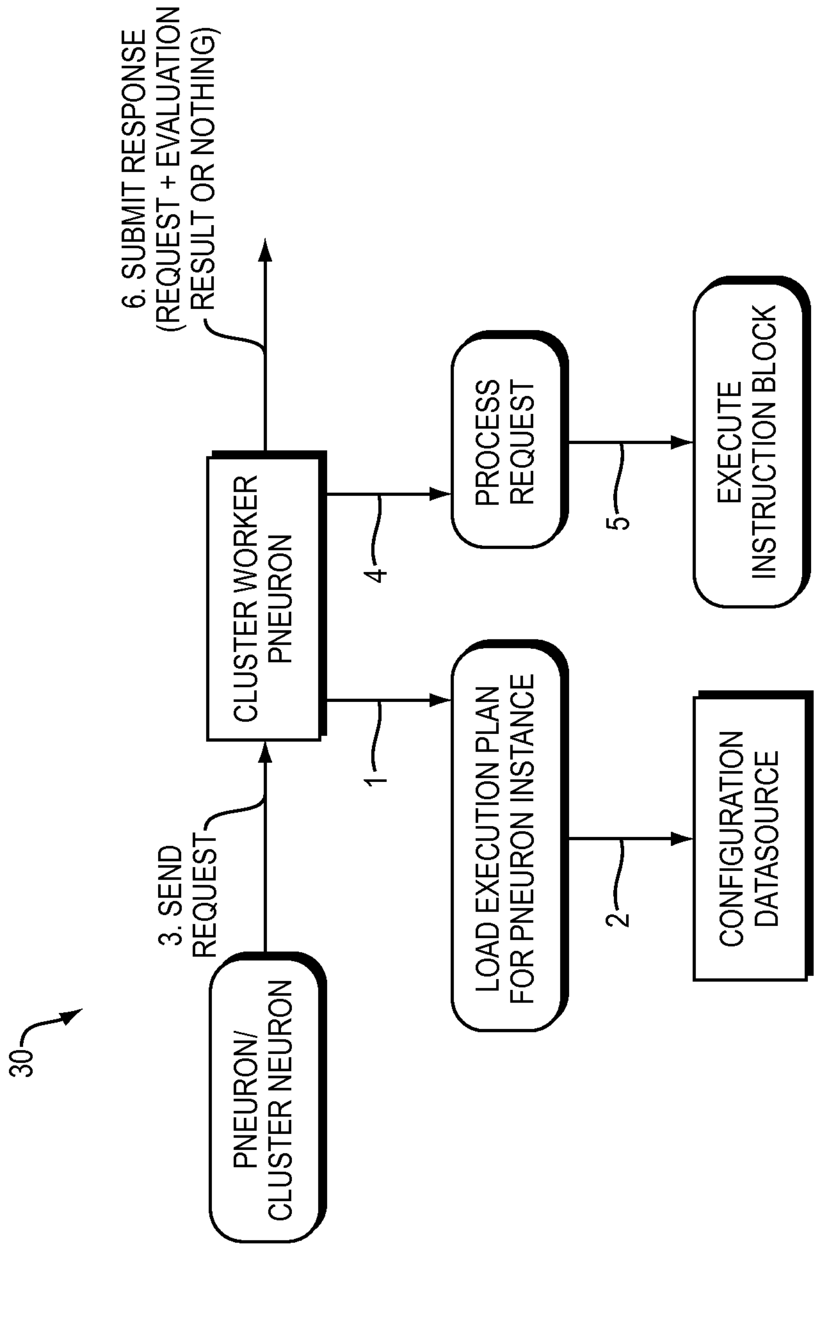
30
PNEURON/
CLUSTER NEURON
3. SEND
REQUEST
CLUSTER WORKER
PNEURON
6. SUBMIT RESPONSE
(REQUEST + EVALUATION
RESULT OR NOTHING)
1
LOAD EXECUTION PLAN
FOR PNEURON INSTANCE
4
PROCESS
REQUEST
2
CONFIGURATION
DATASOURCE
5
EXECUTE
INSTRUCTION BLOCK

FIG. 2

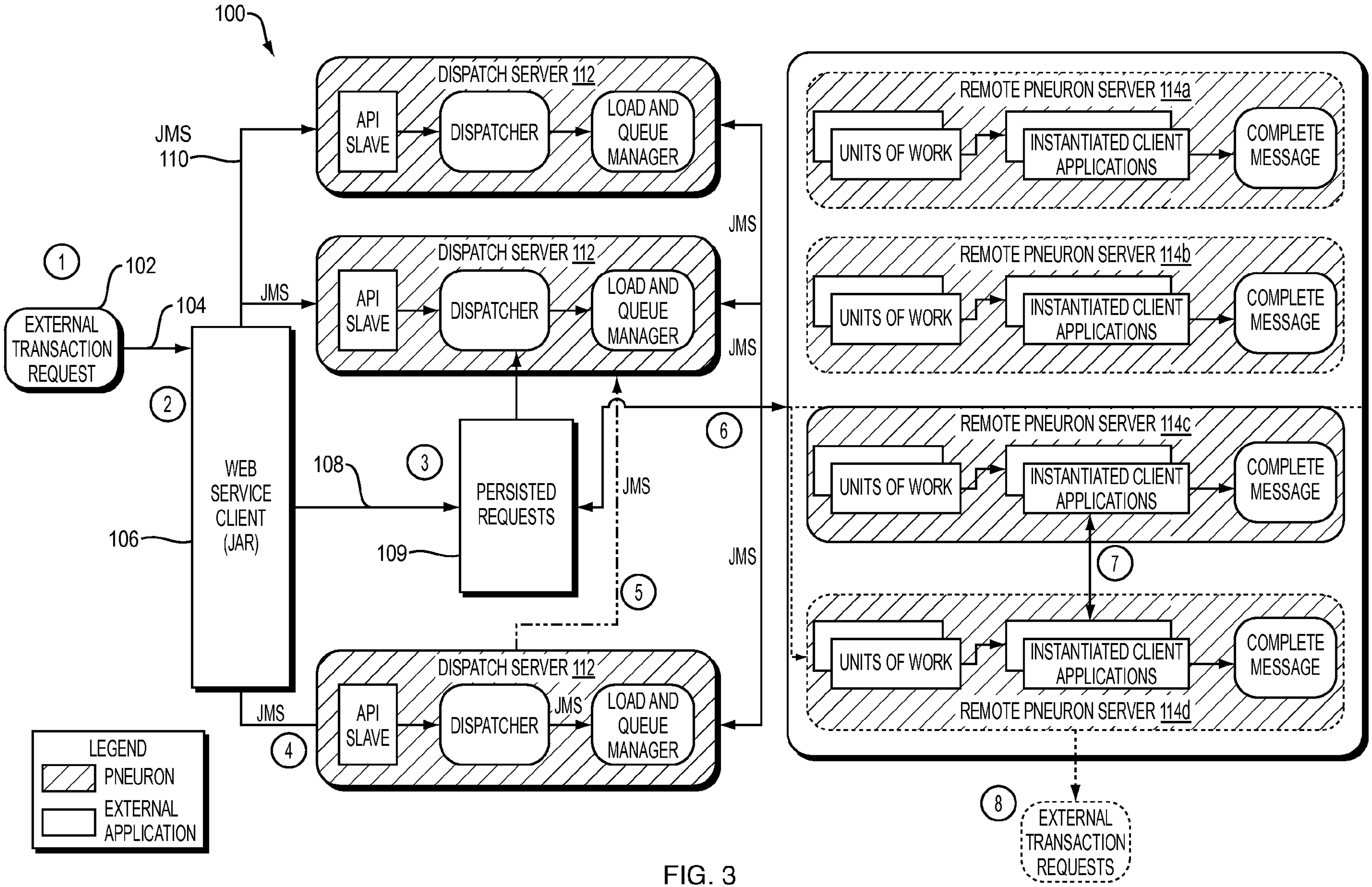
100
1
102
EXTERNAL TRANSACTION REQUEST
104
2
WEB SERVICE CLIENT (JAR)
106
JMS
110
JMS
JMS
4
108
3
PERSISTED REQUESTS
109
JMS
5
6
DISPATCH SERVER 112
API SLAVE
DISPATCHER
LOAD AND QUEUE MANAGER
DISPATCH SERVER 112
API SLAVE
DISPATCHER
LOAD AND QUEUE MANAGER
DISPATCH SERVER 112
API SLAVE
DISPATCHER
JMS
LOAD AND QUEUE MANAGER
JMS
JMS
JMS
REMOTE PNEURON SERVER 114a
UNITS OF WORK
INSTANTIATED CLIENT APPLICATIONS
COMPLETE MESSAGE
REMOTE PNEURON SERVER 114b
UNITS OF WORK
INSTANTIATED CLIENT APPLICATIONS
COMPLETE MESSAGE
REMOTE PNEURON SERVER 114c
UNITS OF WORK
INSTANTIATED CLIENT APPLICATIONS
COMPLETE MESSAGE
7
UNITS OF WORK
INSTANTIATED CLIENT APPLICATIONS
COMPLETE MESSAGE
REMOTE PNEURON SERVER 114d
8
EXTERNAL TRANSACTION REQUESTS
LEGEND
PNEURON
EXTERNAL APPLICATION

FIG. 3

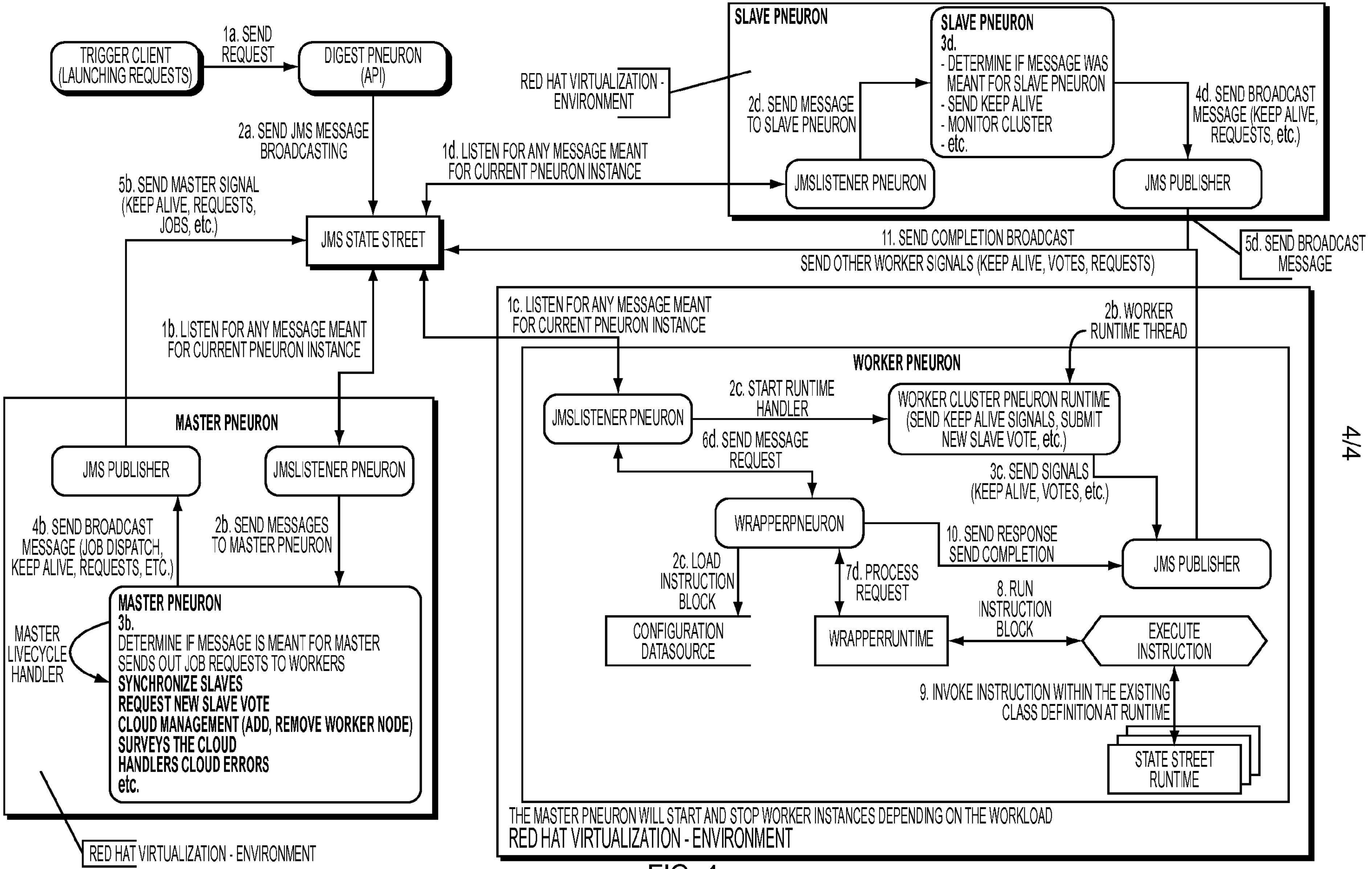
TRIGGER CLIENT (LAUNCHING REQUESTS)
1a. SEND REQUEST
DIGEST PNEURON (API)
2a. SEND JMS MESSAGE BROADCASTING
RED HAT VIRTUALIZATION - ENVIRONMENT
SLAVE PNEURON
SLAVE PNEURON
3d.
- DETERMINE IF MESSAGE WAS MEANT FOR SLAVE PNEURON
- SEND KEEP ALIVE
- MONITOR CLUSTER
- etc.
2d. SEND MESSAGE TO SLAVE PNEURON
4d. SEND BROADCAST MESSAGE (KEEP ALIVE, REQUESTS, etc.)
1d. LISTEN FOR ANY MESSAGE MEANT FOR CURRENT PNEURON INSTANCE
JMSLISTENER PNEURON
JMS PUBLISHER
5b. SEND MASTER SIGNAL (KEEP ALIVE, REQUESTS, JOBS, etc.)
JMS STATE STREET
11. SEND COMPLETION BROADCAST
SEND OTHER WORKER SIGNALS (KEEP ALIVE, VOTES, REQUESTS)
5d. SEND BROADCAST MESSAGE
1b. LISTEN FOR ANY MESSAGE MEANT FOR CURRENT PNEURON INSTANCE
1c. LISTEN FOR ANY MESSAGE MEANT FOR CURRENT PNEURON INSTANCE
2b. WORKER RUNTIME THREAD
WORKER PNEURON
MASTER PNEURON
JMS PUBLISHER
JMSLISTENER PNEURON
4b. SEND BROADCAST MESSAGE (JOB DISPATCH, KEEP ALIVE, REQUESTS, ETC.)
2b. SEND MESSAGES TO MASTER PNEURON
MASTER LIVECYCLE HANDLER
MASTER PNEURON
3b.
DETERMINE IF MESSAGE IS MEANT FOR MASTER
SENDS OUT JOB REQUESTS TO WORKERS
SYNCHRONIZE SLAVES
REQUEST NEW SLAVE VOTE
CLOUD MANAGEMENT (ADD, REMOVE WORKER NODE)
SURVEYS THE CLOUD
HANDLERS CLOUD ERRORS
etc.
RED HAT VIRTUALIZATION - ENVIRONMENT
JMSLISTENER PNEURON
2c. START RUNTIME HANDLER
WORKER CLUSTER PNEURON RUNTIME (SEND KEEP ALIVE SIGNALS, SUBMIT NEW SLAVE VOTE, etc.)
6d. SEND MESSAGE REQUEST
3c. SEND SIGNALS (KEEP ALIVE, VOTES, etc.)
WRAPPERPNEURON
10. SEND RESPONSE SEND COMPLETION
JMS PUBLISHER
2c. LOAD INSTRUCTION BLOCK
7d. PROCESS REQUEST
8. RUN INSTRUCTION BLOCK
CONFIGURATION DATASOURCE
WRAPPERRUNTIME
EXECUTE INSTRUCTION
9. INVOKE INSTRUCTION WITHIN THE EXISTING CLASS DEFINITION AT RUNTIME
STATE STREET RUNTIME
THE MASTER PNEURON WILL START AND STOP WORKER INSTANCES DEPENDING ON THE WORKLOAD
RED HAT VIRTUALIZATION - ENVIRONMENT

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 12/32726

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06F 7/00, G06F 17/30 (2012.01)
USPC - 707/762

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)
IPC(8): G06F 7/00, G06F 17/30 (2012.01)
USPC: 707/762

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 707/999.01,E17.117 | 709/218,220,223,224,228,250 | 706/27 (view search terms below)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWEST(PGPB,USPT,EPAB,JPAB); Google Scholar; wrapper, virtual server, API, application program interface, legacy, cloud, program, software, encapsulate, instance, request, automate, system, computer, electronic, exoskeleton, computerized

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	ZHANG et al. "Migrating Legacy Applications to the Service Cloud" [online] In proceedings of Object Oriented Programming, Systems, Languages and Applications (OOPSLA), 2009: Towards best practice in Cloud Computing. Published 2009. pg. 59-68. [retrieved on 2012-07-01] Retrieved from the Internet <URL: http://www.soacloudbestpratices.com/Towards_best_practices_sammensatt_a5.pdf#page=65 > entire document, especially Abstract; pg. 59-60, 63-64	1-2
Y	US 2003/0058277 A1 (BOWMAN-AMUAH) 27 March 2003 (27.03.2003) entire document, especially Abstract; para [3426], [3440]-[3443], [3454]-[3462]	1-2
A	SNEED. "Encapsulation of legacy software: A technique for reusing legacy software components." [online] In Annals of Software Engineering, Vol. 9, Issue 1-4. Published 2000. pg. 293-313. [retrieved on 2012-07-01] Retrieved from the Internet <URL: http://reversingproject.info/project_repository/reversing_references/pdf/encapsulation_of_legacy_software.pdf > entire document	1-2
A	US 2009/0254572 A1 (REDLICH et al.) 08 October 2009 (08.10.2009) entire document	1-2

□ Further documents are listed in the continuation of Box C. □

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

01 July 2012 (01.07.2012)

Date of mailing of the international search report

17 AUG 2012

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-3201

Authorized officer:

Lee W. Young

PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774