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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 October 2016 (06.10.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/155835 Al

(51) International Patent Classification:
G06F 9/50 (2006.01)

(21) International Application Number:
PCT/EP2015/057344

(22) International Filing Date:
2 April 2015 (02.04.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **TELEFONAKTIEBOLAGET LM ERIC¬SSON (PUBL)** [SE/SE]; SE-164 83 Stockholm (SE).

(72) Inventors: **AELKEN, Joerg;** Spellegasse 20a, B-473 1 Eynatten (BE). **EL KHAYAT, Ibtissam;** Rue Henri Van Der Wielen, 2, B-4690 Glons (BE). **LINDGREN, Tommy;** Torsvikssvangen 35, S-181 34 Lidingo (SE).

(74) Agent: **RÖTHINGER, Rainer;** Wuesthoff & Wuesthoff, Patentanwalte PartG mbB, Schweigerstrasse 2, 81541 Munchen (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

[Continued on nextpage]

(54) Title: TECHNIQUE FOR SCALING AN APPLICATION HAVING A SET OF VIRTUAL MACHINES

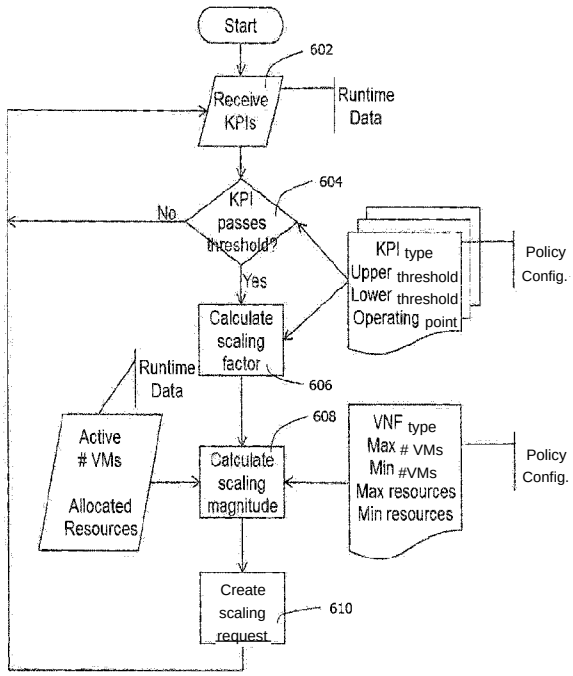

Fig. 6

(57) Abstract: A technique for scaling an application with a set of one or more virtual machines is described. The technique may be performed during runtime of the application and responsive to a determination that a scaling operation is required for the application. That determination can be based on at least one first performance measurement result for the application. A method implementation of the technique comprises calculating a scaling magnitude for the required scaling operation taking into account at least one second performance measurement result for the application. The scaling magnitude is indicative of a resource quantity to be added to or removed from the application. The method further comprises triggering generation of a scaling request. The scaling request is directed at a scaling of the application on the basis of the calculated scaling magnitude.

WO 2016/155835 A1

Published:

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

Technique for scaling an application having a set of virtual machines

Technical Field

The present disclosure generally relates to cloud computing. In particular, a technique for scaling an application having a set of virtual machines is presented. The technique may be practiced in the form of a method, a computer program, an arrangement (e.g., an apparatus or node) and a system.

Background

The workgroup on Network Function Virtualization (NFV) within the European Telecommunications Standard Institute (ETSI) has published recommendations on lifecyde management of Virtualized Network Functions (VNF) in their framework ETSI GS NFV-MAN 001, VI. 1.1 (2014-12) entitled NFV Management and Orchestration (MANO). One particular aspect of VNF lifecyde management is VNF elasticity via scaling processes.

The VNF scaling processes defined in the ETSI VNF MANO framework include scale-out/-in and scale-up/-down operations (see, e.g., section B.4.4.). Scale-out/-in operations are directed at changing the capacity of a VNF by means of adding or removing Virtual Machines (VMs). Scale-up/-down operations encompass changing the capacity of a VNF by means of adding or removing infrastructure resources (e.g., in terms of computing, network and storage resources) to or from existing VMs.

The execution of a VNF scaling process is triggered by a system entity detecting the need for a capacity increase or decrease via monitoring Key Performance Indicators (KPIs) of the VNF or of its underlying infrastructure. The behavior of the detection entity is usually configured by means of policies. Current policies to configure the decisions for VNF scaling are based on thresholds for certain KPI types (see, e.g., section B.4.4.3 of the ETSI VNF MANO framework). The monitoring of the KPIs and their comparison against the thresholds enable the detection of threshold passing. For example, in the event of a KPI relaxing below a configured threshold, the need for a capacity decrease is detected and execution of a scale-in or scale-down operation will be triggered.

It has been found that the VNF scaling process as defined in the ETSI VNF MANO framework is not optimal in many aspects. For example, the speed of convergence of the scaling process is strongly varying and difficult to predict. As such, the scaling process does not exhibit a deterministic behavior. Similar problems are also encountered for applications with virtual machines that conform to MANO standards different from the ETSI VNF MANO framework.

Summary

Accordingly, there is a need for a technique that permits a more efficient scaling of applications having a set of one or more virtual machines.

According to a first aspect, a method of scaling an application having a set of one or more virtual machines is provided. The steps of the method are performed during runtime of the application and responsive to a determination that the scaling operation is required for the application, wherein the determination is based on at least one first performance measurement result obtained for the application. In detail, the method comprised calculating a scaling magnitude for the required scaling operation taking into account at least one second performance measurement result obtained for the application, wherein the scaling magnitude is indicative of a resource quantity to be added to or removed from the application. The method further comprises triggering generation of a scaling request, wherein the scaling request is directed at a scaling of the application on the basis of the calculated scaling magnitude.

In certain variations, the method may also comprise determining, prior to the calculation of the scaling magnitude, that a scaling operation is actually required for the application. That determination can be based on the same performance measurement result that will also be taken into account for the calculation of the scaling magnitude or on a different performance measurement result.

In one variant, an operating target is defined for a performance indicator underlying the second performance measurement result. The performance indicator may, for example, be a load parameter of the application, that is measured to obtain the second performance measurement result. The operating target may generally be an operating point or an operating range for a given performance indicator.
The scaling magnitude may be calculated based on a present (i.e., current) or expected relationship between the performance indicator and the operating target.

An expected relationship between the performance indicator and the operating target may, for example, be derived by extrapolating one, two or more second performance measurement results obtained for the same performance indicator at different points in time.

A scaling factor may be determined from the present or expected relationship between the performance indicator and the operating target. The relationship between the operating target and the performance indicator may be expressed in various ways. As an example, the relationship may be defined as a current or expected deviation of the performance indicator from the operating target.

The scaling factor may generally be taken into account for calculating the scaling magnitude. As an example, the scaling magnitude may be calculated from the scaling factor and a resource quantity presently allocated to the application. In certain variants, the scaling magnitude may be determined by multiplying the presently allocated resource quantity with the scaling factor. The result of this multiplication may be processed further (e.g., offset) to obtain the scaling magnitude.

In certain variants the scaling magnitude is calculated taking into account multiple second performance measurement results obtained for multiple performance indicators. As an example, for each performance indicator a dedicated second measurement result may be obtained. In such a case a dedicated operating target may be defined for each performance indicator. The scaling magnitude may then be calculated based on present or expected relationships between the performance indicators and the associated operating targets.

There may exist a known correlation between the second performance measurement result and the resource quantity to be added to or removed from the application. As an example, the correlation may be a functional (e.g., essentially linear) relationship or a mapping. The correlation may have been determined prior to runtime of the application (e.g., via an empirical approach that can be based on measurements).

The known correlation may be taken into account in the calculation of the scaling magnitude. As an example, the scaling magnitude may be determined from the correlation and the relationship between the operating target and the performance indicator.

The second performance measurement result is in one example indicative of a system performance of the application The second measurement result may thus have been obtained by aggregating individual performance measurements over the set of virtualized machines. As an example, for each individual virtual machine in the set a dedicated individual performance measurement may be performed. The resulting individual performance measurement results can be aggregated (e.g., added, averaged, etc.) so as to obtain the ("final") second performance measurement result that will be taken into account in the scaling magnitude calculation.

At least one of the first measurement result and the second measurement result may be indicative of a load of the application. Additionally, or in the alternative, at least one of the first measurement result and the second measurement result may be independent from the number of virtual machines associated with the application. As explained above, averaging of individual performance measurement results obtained for each individual virtual machine could be applied to that end.

The determination that a scaling cooperation is required and the calculation of the scaling magnitude may be performed on the basis of one and the same performance measurement result or set of performance measurement results. As such, the (first) measurement result underlying the determination that the scaling operation is required may be used as the (second) measurement result that is taken into account upon calculating the scaling magnitude.

As said, the method presented herein may further comprise determining that a scaling operation is required. That determination may be performed in various ways, for example by subjecting the first performance measurement result to at least one threshold decision. In some variants, a lower threshold and an upper threshold for the first performance measurement result may be defined. The determination may in certain variants also be performed based on the operating target for the performance indicator. As an example, it may be determined that a scaling operation is required upon detection of a predefined deviation of the first performance measurement result from the operating target.

The operating target (e.g., the operating point or operating range) for the performance indicator may lie between the lower threshold and the upper threshold. In other variants, the operating target may at least partially lie below the lower threshold or above the upper threshold. There may exist a predefined relationship

between the operating target for the performance indicator on the one hand and at least one of the lower threshold and the upper threshold on the other.

The method may further comprise verifying the calculated scaling magnitude. Moreover, the method may optionally comprise adjusting the calculated scaling magnitude dependent on a result of the verification. The scheduling request may be triggered to be generated such that it is indicative of the adjusted scaling magnitude.

The verification of the calculated scaling magnitude may be performed in various ways, for example by comparing the calculated scaling magnitude with at least one configuration parameter. A threshold decision may be applied in this regard.

The at least one configuration parameter may be selected from a parameter set comprising a maximum number of allowed virtual machines for the application, a minimum number of allowed virtual machines for the application, a maximum amount of allowed infrastructure resources for an individual virtual machine, and a minimum amount of allowed infrastructure resources for an individual virtual machine. As such, the resource quantity may be indicative of a number of virtual machines to be added to or removed from the application. Alternatively, or in addition, the resource quantity may be indicative of infrastructure resources for the virtual machines to be added to or removed from (e.g., per virtual machine) the application.

Also provided is a computer program product comprising program code portions for performing the steps of any of the methods and method steps presented herein when the computer program product is executed by at least one computing device (e.g., a processor or a distributed set of processors). The computer program product may be stored on a computer-readable recording medium, such as a semiconductor memory, a CD-ROM, DVD, and so on.

According to a still further aspect, an arrangement configured to trigger scaling of an application having a set of one or more virtual machines is presented. The arrangement comprises at least one processor configured to perform dedicated operations during runtime of the application and responsive to a determination that a scaling operation is required for the application, wherein the determination is based on at least one first performance measurement result obtained for the application. Specifically, the processor is configured to calculate a scaling magnitude for the required scaling operation taking into account at least one second performance

measurement result obtained for the application, wherein the scaling magnitude is indicative of a resource quantity to be added to or removed from the application. The processor is further configured to trigger generation of a scaling request, wherein the scaling request is directed at a scaling of the application based on the calculated scaling magnitude.

The application may be configured as a VNF. Moreover, the arrangement may be configured as a VNF Manager (VNFM). The VNF and VNFM may conform to ETSI GS NFV-MAN 001, VI. 1.1 (2014-12). It should be noted that the arrangement could also be configured in any other manner and is thus not limited to being implemented in a telecommunications scenario.

The arrangement may generally be configured to perform any of the methods and method steps presented herein. Moreover, the arrangement may be configured as an apparatus, a network node or a set of network nodes.

Also provided is a system comprising the arrangement presented herein and the application having the set of one or more virtual machines. The system may belong to a telecommunications cloud system. The telecommunications cloud system may further comprise at least one of a Radio Base Station (RBS) function, an Evolved Packet Core (EPC) function, an Internet Protocol Multimedia Subsystem (IMS) core function, and one or more other functions running on the set of virtual machines.

Brief Description of the Drawings

Further details, aspects and advantages of the present disclosure will become apparent from the following description of exemplary embodiments and the accompanying drawings, wherein:

Fig. 1 schematically illustrates an embodiment of a telecommunications cloud system in which the present disclosure may be implemented;

Fig. 2 schematically illustrates an embodiment of a triggering arrangement;

Fig. 3 illustrates a flow diagram of a method embodiment performed by the triggering arrangement of Fig. 2;

Fig. 4 schematically illustrates an embodiment of an application underlying a scaling operation;

Fig. 5 schematically illustrates a signalling diagram into which embodiments of the present disclosure may be integrated;

Fig. 6 illustrates a further flow diagram of a method embodiment;

Fig. 7 schematically illustrates an embodiment of KPI aggregation; and

Fig. 8 illustrates a diagram underlying an exemplary scaling scenario.

Detailed Description

In the following description, for purposes of explanation and not limitation, specific details are set forth, such as specific network nodes, network configurations, communication protocols, and so on, in order to provide a thorough understanding of the present disclosure. It will be apparent to one skilled in the art that the present disclosure may be practiced in other embodiments that depart from these specific details. For example, while the following embodiments will partially be described in connection with exemplary cloud architectures and an exemplary ETSI recommendation, it will be appreciated that the present disclosure may also be practiced in connection with other cloud architectures and other cloud management and orchestration approaches. It will also be appreciated that the present disclosure is not limited to be applied in connection with telecommunications systems. Rather, the present disclosure could, for example, also be implemented in connection with online sales or other enterprise applications.

Those skilled in the art will further appreciate that the steps, services and functions explained herein below may be implemented using individual hardware circuitry, using software functioning in conjunction with a programmed micro-processor or general purpose computer, using one or more Application Specific Integrated Circuits (ASICs) and/or using one or more Digital Signal Processors (DSPs). It will also be appreciated that when the present disclosure is described in terms of a method, it may also be embodied in one or more processors and one or more memories coupled to the one or more processors, wherein the one or more memories are encoded with one or more programs that perform the steps, services and functions disclosed herein when executed by the one or more processors.

Fig. 1 illustrates an embodiment of a possible cloud architecture 100 of a 5th Generation (5G) telecommunications network system in which the present disclosure may be implemented. The cloud architecture 100 logically separates network functions potentially running on virtualized hardware (functional layer 110 in Fig. 1) from the infra-structure or hardware layer 120 containing the physical nodes in the 5G network system.

The functional layer 110 contains the functions (Network Functions (NF) and Dedicated Functions (DF)) performed by the 5G network system including tasks like mobility, security, routing, baseband processing, etc. Many but not necessarily all of these NFs will be performed by software running on virtualized hardware. Some of these NFs running on virtualized hardware will utilize Application Program Interfaces (API) provided by an execution environment to be able to control functionalities executed in hardware such as Service Defined Network (SDN) switches, hardware acceleration and so on.

Since at least some of these NFs are virtualized (VNFs), they are not tied to a specific hardware node. That is, they can be executed in different places within the network system depending on the given deployment scenario and requirements. This approach makes it possible to, for instance, distribute in a flexible way gateway functionalities closer to radio access nodes 130 when needed for particular services, while supporting more centralized gateways for other services. In theory this also makes it possible to dynamically re-configure the network system based on ongoing services or load. However, in the 2020 time frame it is still expected that time critical functions such as baseband processing today performed by dedicated hardware in the access nodes 130 (implementing DFs) will in most cases continue to do so.

The infrastructure (hardware) layer 120 of the cloud architecture 100 contains radio nodes including user terminals (also called User Equipment, UEs), relay nodes (including wireless MTC-gateways or self-backhauled nodes) and one or more RANs 140 with the access nodes 130. In Fig. 1, the access nodes 130 are separated in antenna, Radio Unit (RU) and Digital Unit (DU). Further, the infrastructure layer 120 comprises network nodes including processing, switches/routers and storage nodes 150 and one or more data centers 160. The nodes 150 may, for example, be configured to host EPC services or functions.

The cloud model underlying cloud architectures, such as the architecture 100 shown in Fig. 1, can be divided into four layers: the hardware layer (1), the infrastructure layer (2), the platform layer (3) and the application layer (4). Each higher layer builds on top of the features and services provided by the lower layers.

The hardware layer typically refers to the data center(s) 160 and other core infrastructure nodes 150 (see Fig. 1). The infrastructure is offered as infrastructure-as-a-service (IaaS) at layer 2. Then, at the layer 3, the platform layer, high-level platforms and environments are provided to develop software or services often refered to as platform-as-a-service (PaaS). These platforms usually take the form of operating systems and/or software frameworks. The point is to shield from dealing with the underlying complexities of the infrastructure entities such as Virtual Machine (VM) containers and raw storage blocks.

Finally, at the application layer, there are installed generally one or more service provider applications providing, in the present embodiment, telecommunications services and, in more general realizations, business applications, web services, multimedia and gaming services. All of these qualify as software-as-a-service (SaaS) in the cloud paradigm terminology. The scaling approach presented herein may be performed in relation to an application residing on the application layer.

Due to the high availability requirements in the cloud architecture 100 of Fig. 1, it is critical to develop techniques for service assurance to fulfill those requirements, but also many other requirements. This objective typically involves continuous monitoring of relevant KPIs relating, for example, to a specific SLA for a given service (e.g., an RBS service within the RAN 140), analyzing the data for finding abnormal trends and anomalies and triggering the suitable cloud orchestration actions, such as scaling operations, in case of any violations.

For example, during peak hours extra capacities are required by critical applications supporting, for example, the RAN 140. On the other hand, during off-peak hours the capacity requirement may drastically relax. To cope with these and other load fluctuations in the network system of Fig. 1, efficient scaling solutions as presented herein may be applied.

Fig. 2 illustrates an embodiment of a triggering arrangement 20 configured to trigger scaling of an application having a set of one or more VMs, such as an application within the network system shown in Fig. 1 or any other application. The arrangement

20 can be configured as a network node or network function, or as a distributed set of network nodes and network functions. The arrangement 20 could also (fully or partially) reside on a UE.

As shown in Fig. 2, the triggering arrangement 20 comprises at least one processor 22, at least one interface 24 and at least one memory 26. The at least one processor 22 is configured to perform processing operations under control of program code stored in the memory 26. The one or more interfaces 24 are configured as software and/or hardware interfaces. Specifically, the one or more interfaces 24 may be configured to receive and send data and control signaling. In certain variants of the present disclosure, the one or more interfaces 24 are configured to receive performance measurement results and to send scaling requests or control signaling that triggers generation of such scaling requests.

An exemplary mode of operation of the triggering arrangement 20 illustrated in Fig. 2 will now be described with reference to the flow diagram illustrated in Fig. 3. The operation targets at triggering scaling of an application having a set of one or more VMs.

In an (optional) initial step 302, the at least one processor 22 of the triggering arrangement 20, or any other entity, determines that a scaling operation is required for the application with the one or more VMs. That determination may be based on at least one first performance measurement result obtained for the application. The first performance measurement result may have been received by the triggering arrangement 20 via the one or more interfaces 24. Step 302 is performed during runtime of the application.

The determining step 302 may include subjecting the first performance measurement result to one or more threshold decisions. Specifically, a lower threshold and an upper threshold may be defined (e.g., in the memory 26). It may thus be determined in step 302 that a scaling operation is required if the first performance measurement result exceeds the upper threshold or falls below the lower threshold. Alternatively, the determination in step 302 whether a scaling operation is required could also be based on a deviation of the first performance measurement result from a predefined operating target for a performance indicator. Details regarding the operating target will be described below.

Responsive to the determination in step 302 that a scaling operation is required, the at least one processor 22 calculates a scaling magnitude for the required scaling operation in step 304. The calculation in step 304 takes into account at least one second performance measurement result for the application. The second performance measurement result may be identical or different from the first performance measurement result processed in step 302. The second performance measurement result may also have been received via one or more interfaces 24.

The scaling magnitude calculated in step 304 is indicative of a resource quantity to be added to or removed from the application. As an example, the resource quantity may be indicative of a number of VMs to be added to or removed from the application. As a further example, the resource quantity may be indicative of an amount of infrastructure resources (e.g., in terms of one or more of computing, storage and networking resources) to be added to or removed from one or more of the VMs in the set.

In an optional step not depicted in Fig. 3, the scaling magnitude calculated in step 304 may be verified by comparing the calculated scaling magnitude with at least one configuration parameter, such as one or more of a maximum number of allowed VMs for the application, a minimum number of allowed VMs for the application, a maximum amount of allowed infrastructure resources for an individual VM, and a minimum amount of allowed infrastructure resources for an individual VM. Based on the result of the comparison, the scaling magnitude calculated in step 304 can be adjusted so as to meet the one or more configuration parameters.

In a further step 306, generation of a scaling request is triggered by the triggering arrangement 20. The scaling request generated in response to the triggering operation is directed at a scaling of the application on the basis of the calculated scaling magnitude. Step 306 encompasses the case where the calculated scaling magnitude is adjusted responsive to its verification (as the adjusted scaling magnitude will still be based on the scaling magnitude calculated in step 304).

Responsive to the triggering step 306, the scaling request will either be generated locally within the triggering arrangement 20 or by any other entity. As such, the triggering arrangement 20 may send via the one or more interfaces 24 either a triggering event for the scaling request or the scaling request as such to another entity in charge of actually scaling the application, such as a cloud management entity. Fig. 4 schematically illustrates such a cloud management entity 40 in control

of a virtualized application, or simply application, 42. As shown in Figure 4, the application 42 comprises multiple VMs 46. The cloud management entity 40 and the application 42 may belong to the functional layer 110 of the cloud architecture 100 shown in Fig. 1.

The cloud management entity 40 will receive the scaling request or the triggering event for generation of a scaling request either directly from the triggering arrangement 20 or from an entity located between the triggering arrangement 20 and the cloud management entity 40, see Fig. 4. The scaling request or the triggering event will comprise the scaling magnitude as calculated in step 304 that has, optionally, been adjusted responsive to the verification step.

Responsive to receipt of the scaling request or the triggering event, the cloud management entity 40 adds or removes one or more VMs 46 from the application 42 depending on the indicated scaling magnitude. Alternatively, or in addition, the cloud management entity 40 adds or removes infrastructure resources to or from one or more of the VMs 46 dependent on the indicated scaling magnitude.

A VM 46 may generally be constituted by a (virtualized) computing resource. Thus, creation or generation of a VM 46 may refer to deployment or allocation of the associated computing resource. To each computing resource, networking resources and storage resources can be added (e.g., associated, allocated or connected) on demand. Different technologies exist to allocate computing resources and exposed them as VMs 46. Such technologies include a hypervisor as hardware abstraction layer, containers (e.g., Linux containers), PaaS frameworks, and a so-called bare metal virtualization. In the ETSI Framework, the term is used to designate a virualized application 42. A deployed VNF typically consists of multiple instances of one or more (typically different) VM types, where each VM type runs its own, dedicated function.

In certain variants, the calculation step 304 in Fig. 3 may take into account an operating target (e.g., an operating point or operating range) defined for a performance indicator underlying the second performance measurement result that is taken into account in the scaling magnitude calculation. As such, the scaling magnitude may be calculated in step 304 based on a present or expected relationship between the performance indicator (for which the second performance measurement result has obtained) and the operating target. In case multiple second performance measurement results (e.g., for different performance indicators) are

taken into account upon calculating the scaling magnitude in step 304, for each performance indicator a dedicated operating target may be defined.

The operating target and related parameters may be specified in different ways and may be stored in the memory 26 for being accessed by the one or more processors 22 of the triggering arrangement 20 (see Figure 2). As an example, an operating target may be a dedicated performance indicator target point or target range of a predefined scaling policy configuration. Alternatively, or in addition, the operating target may be calculated from one or more threshold values that are analyzed in connection with determining whether or not there exists requirement for a scaling operation (see step 302 in Fig. 3). As an example, a predefined scaling policy configuration stored in the memory 26 may define offset values or any functional relationship to be applied to the one or more threshold values to calculate the one or more operating targets. In certain variants, the predefined policy configuration could also define an operating range for each performance indicator that is taken into account in the calculation step 304.

The scaling magnitude may be calculated based on a present or expected relationship between the respective performance indicator and the respective operating target. As an example, the present relationship may simply be determined by analyzing a deviation of the (current) second performance measurement result from the operating target. An expected relationship may be determined by extrapolating a number of (previous) second performance measurement results into the future and by analyzing a deviation of the extrapolated value from the operating target.

In certain variants, a scaling factor may be determined from the present or expected relationship between the performance indicator and the operating target. In such a case the scaling magnitude may be calculated from the scaling factor and a resource quantity presently allocated to the application (e.g., using a multiplication operation).

For a particular performance indicator, the scaling factor may be determined from the associated operating target (e.g., as stored in the memory 26 as part of a predefined scaling policy configuration). Alternatively, or in addition, the scaling factor may be determined from the present or expected relationship between the performance indicator and the operating target.

In an exemplary implementation compliant, for example, with the ETSI framework, the expected relationship (e.g., deviation) is used for scale-up and scale-out operations. Extrapolation in connection with scale-down and scale-in may be used only if a reaction time for the scale-down/scale-in operation is very slow and the point of extrapolation is not further than the time to complete the associated scaling action. By specifying the operating target for each KPI of an application 42 for each of a scale-out, scale-in, scale-up and scale-down operation individually, an operator can configure the desired behavior of the application 42 concerning the desired load and resource utilization.

In certain scenarios, a dedicated operating target for each performance indicator (e.g., KPI) of interest and for each configured scaling operation (e.g., one of more of scale-out, scale-in, scale-up and scale-down) may be provided. The one or more operating targets (or parameters suitable to derive the one or more operating targets) may be stored in the memory 26 for use by the one or more processors 22 of the triggering arrangement 20 (see Fig. 2). The corresponding parameter set for deriving an operating target may include one or more of an operating point and a maximum permissible deviation relative thereto (i.e., to define an operating range), minimum and maximum values of the operating range (e.g., relative to one or more thresholds utilized in the determination step 302 of Fig. 3) and a simple target operating point. Based on these configuration parameters (that define a particular scaling policy) and runtime data (e.g., performance measurement results obtained for one or more KPIs), the scaling magnitude can then be calculated in step 304 as generally explained above. Further configuration data stored in the memory 26 and used for determining the scaling magnitude may include the maximum/minimum number of allowed VMs 46 and of allowed infrastructure resource for an individual VM 46 as discussed above.

The runtime data in terms of the (first and second) performance measurement results may continuously be received at runtime of the application 42. Further, the calculation of the scaling magnitude in step 304 may also take into account runtime information on the number of active VMs 46 in the application 42 and the actual amount of allocated infrastructure resources per application 42 or VM 46 in the application 42.

In the following, the embodiments described above will exemplarily be put in the larger context of ETSI GS MFV-MAN 001, VI. 1.1 (2014-12). It will appreciated that

the following details of exemplary scaling approaches could likewise be applied in connection with other cloud management and orchestration approaches.

Fig. 5 illustrates an embodiment of a signaling diagram in which the present disclosure can be embedded. The signaling diagram is based on Fig. B.13 of the ETSI framework and shows the signaling between the following components: Element Management (EM), VNFM, NFV Orchestrator (NFVO), and Virtualized Infrastructure Manager (VIM).

In step 1, the VNFM is continuously informed during runtime of the VNF (e.g., in the form of the application 42 illustrated in Fig. 4) about performance measurement results pertaining to one or more KPIs. The VNFM collects the measurement results and detects in step 2 the requirement for a scaling operation and also calculates a scaling magnitude indicative of a required resource quantity (e.g., as generally described with reference to Fig. 3 above). The VNFM may detect a requirement for a scaling operation from a capacity shortage in the VNF that requires an expansion (e.g., an addition of resources to the VNF).

The VNFM then generates a scaling request comprising the calculated scaling magnitude and sends the scaling request in step 3 to the NFVO for VNF expansion using the operation Grant Lifecycle Operation of the VNF Lifecycle Operation Granting interface. In step 4, the NFVO takes a scaling decision and checks the scaling magnitude in the resource request received from the VNFM against its capacity database for free resource availability. The remaining steps 5 to 15 in the signaling diagram of Fig. 5 are generally in line with the ETSI framework and will therefore not be described in greater detail herein.

Fig. 6 illustrates a further flow diagram of a method embodiment that may be performed by the triggering arrangement in Fig. 2, and, in particular by the VNFM (e.g., in step 2 of Fig. 5).

Initially, in step 602, runtime data are received. The runtime data include performance measurement results for multiple KPIs.

The measurement results received in step 602 have been previously aggregated as generally illustrated in Fig. 7. That is, each VM 46 in the application 42 reports its local performance measurement result obtained for a particular KPI to a KPI aggregator 70. The KPI aggregator 70 aggregates the reported measurement results

such that the resulting aggregated measurement result is independent of the number of reporting VMs 46. As an example, the KPI aggregator may apply an averaging procedure.

Then, in step 604, the individual (aggregated) measurement results for the various KPIs are individually subjected to a threshold decision to determine the requirement of a scaling operation (in accordance with step 302 of Fig. 3). For each KPI a lower threshold and an upper threshold may be defined as explained above. In case no threshold violation is detected in step 604, the method loops back to step 602.

In case it is determined in step 604 that for at least one KPI the associated upper threshold or lower threshold is passed, the method proceeds to step 606. In step 606 a scaling factor is calculated. There exist various algorithmic options for calculating the scaling factor.

One exemplary algorithm assumes a linear functional relationship between the performance measurement results and the number of VMs 46 or allocated infrastructure resources of the VMs 46 within the application 42. For each KPI type value KPI, passing an associated threshold value (as determined in step 604), a KPI type specific scaling factor SF, can be calculated as follows:

$$SF_i = \frac{absiKPIi \quad OPi)}{\text{OPi.,}}$$

wherein OP, denotes the operating point value for this KPI type. In a simple form, the scaling factor SF is set to the maximum of the individual scaling factors SF, in case of an exemplary scale-out operation. That is

$$\mathbf{SF} = \max(SFi)$$

In its simple form, the scaling factor SF is set to the minimum of the individual scaling factors SF, in case of a scale-in operation. In a similar manner, scale-up and scale-down operations are handled. In general, different weights could be given to the different KPIs. The scaling factor SF may then be impacted by these weights.

Of course, other algorithms could be used as well for calculating the scaling factor. Generally, there will often be a way to define one or more KPIs that scale linearly with the number of VMs 46 or the amount of allocated resources. Another way of

defining the correlation of the performance measurement results and the number of VMs 46 or allocated infrastructure resources comprises running the application 42 with different numbers of VMs or different amounts of allocated infrastructure resources and different loads, and capturing the correlation between the performance measurement results and the required number of VMs 46 or required amount of infrastructure resources. These processes are performed prior to runtime of the application 42, for example, during an installation phase of the system. As a result, a functional relationship or a mapping between the performance measurement results and the required resource quantity to be added or to be removed can be determined.

From step 606, the process illustrated in Fig. 6 moves to step 608 and the calculation of the scaling magnitude. Steps 606 and 608 generally correspond to step 304 in Fig. 3.

In detail, the scaling factor determined in step 606 is multiplied with the number of active VMs 46 or the amount of allocated infrastructure resources. The result of this multiplication is rounded up or down to an integer value so as to obtain the scaling magnitude. The scaling magnitude will thus be indicative of the particular resource quantity to be added or removed from the application.

In order to avoid exceeding or falling below a configured maximum or minimum size of the application 42 in terms of the number of VMs 46 or amount of infrastructure resources, the scaling magnitude calculated in step 608 can be verified (not shown in Fig. 6). The verification process may include the calculation of the potential new number of VMs 46 or amount of infrastructure resources taking into account the currently deployed resource quantity and adding or removing resource quantity in accordance with scaling magnitude to or from the currently deployed resource quantity. If the resulting number of VMs 46 or amount of infrastructure resources violates the corresponding maximum or minimum conditions, the scaling magnitude will be adjusted accordingly (e.g., limited to the particular maximum or minimum value that is passed).

Then, in step 610, the scaling request is generated that includes the calculated and, potentially, adjusted scaling magnitude. The processing of the scaling request may then be performed as generally illustrated in Fig. 5 (steps 4 to 15), and the method loops back to step 602. In this regard, a "protection period" in which no further scaling request is or can be triggered might be implemented.

In the following, an actual example of a scale-out operation will be described with reference to the exemplary diagram shown in Fig. 8. Fig. 8 illustrates, for a particular KPI, an upper threshold and a lower threshold for a performance measurement result (KPI runtime value) that is subjected to the determination step 302 in Fig. 3 or 604 in Fig. 6. Also, a dedicated operating point underlying the calculation of the scaling magnitude in step 304 or 608 is illustrated.

For the purpose of the following example, a "VM load KPI" is defined as the arrival rate of requests in an individual VM 46 divided by the maximum number of requests the individual VM 46 can handle:

$$Load_{VM} = \frac{rate\,of\ coming\ requests}{max\ Req}$$

The arrival rate will be measured over a configurable time interval. The maximum number of requests one VM 46 can handle is supposed to be a known value (e.g., a predefined number).

To determine the system performance of the set of VMs 46 defining a particular application 42 (see Fig. 4), the scaling factor SF is derived as an average of the KPI values (i.e., measurement results) obtained for a number num_VM of individual VMs 46 in the application 42. It will be assumed here that all VMs are of the same "size" (i.e., can handle the same load in terms of incoming requests). Then, a "load system KPI" can be expressed as:

$$\mathbf{Load\_sys} = \left(\sum_{k=0}^{num_{VM}} \mathbf{Load_{VM_k}}\right)/num\_\{VM\},$$

where $\mathbf{Load_{VM\_k}}$ is the load of the k'th VM 46.

In the example illustrated in Fig. 8 we consider that:

- the upper threshold (as applied, e.g., in steps 302 and 604) is set to 80%

- the lower threshold (as applied, e.g., in steps 302 and 604) is set to 40%

- the operating point (as applied, e.g., in steps 304 and 608) is set to 70%

Let us suppose that we have a system with three VMs 46 (i.e., n = 3 in Fig. 7). Each VM 46 is capable of handling a maximum of 100 requests/sec. Let us also suppose that, just before an increase of load, all the VMs 46 are handling 60 requests/sec each. Let us further suppose that there will be a load increase of 66% (i.e, 120 requests/sec coming more into the application 42). In the scenario of Fig. 7, Vml ends up with 110 requests/sec, VM2 with 90 requests/sec and VM3 with 100 requests/sec (there is no control on the load balancing in the present example). Thus, Vml, Vm2 and Vm3 will send load KPI values of 1.1, 0.9, and 1, respectively, to the KPI aggregator 70.

The KPI aggregator 70, based on these three values, calculates the system load KPI value as defined above to equal:

$$Load\_sys = 1.$$

It is this value that enters as measurement result step 602 in Fig. 6 (see also Fig. 8). It will thus be determined in step 604 that the upper threshold (i.e., 80 requests/sec) is passed. The scaling factor is then determined in step 606 to equal:

$$SF= (1- 0.7)/0.7=0.43$$

The number of VMs 46 to be added to the application 42 (i.e., the scaling magnitude) can then be calculated in step 608 by multiplying that scaling factor SF with the current number of VMs 46 utilized by the application 42:

$$Ceil(SF*current\_number\_VM)=Ceil(0.43*3)=2$$

This means that the scaling request sent in step 610 will indicate that two VMs 46 have to be newly added to the application 42.

As has become apparent from the above description of exemplary embodiments, the behavior of applications with one or more VMs can be controlled more deterministic in terms of load and resource utilization. In certain variants, an operator is able to specify desired operating targets for the application. Moreover, the capacity of an application can be adapted faster to the current load, and will also converge faster to a desired operating target.

The present disclosure may, of course, be carried out in other ways than those specifically set forth herein without departing from the scope of the claims appended hereto. Thus, the present embodiments are to be considered in all respects as illustrative and not restrictive, and all changes coming within the scope the appended claims are intended to be embraced therein.

Claims

1. A method of triggering scaling of an application (42) having a set of one or more virtual machines (46), wherein the method comprises the following steps performed during runtime of the application and responsive to a determination (302; 604) that a scaling operation is required for the application, wherein the determination is based on at least one first performance measurement result obtained for the application:

calculating (304; 606, 608) a scaling magnitude for the required scaling operation taking into account at least one second performance measurement result obtained for the application, wherein the scaling magnitude is indicative of a resource quantity to be added to or removed from the application; and

triggering generation (306; 610) of a scaling request, wherein the scaling request is directed at a scaling of the application on the basis of the calculated scaling magnitude.

2. The method of claim 1, wherein

an operating target is defined for a performance indicator underlying the second performance measurement result; and wherein

the scaling magnitude is calculated based on a present or expected relationship between the performance indicator and the operating target.

3. The method of claim 2, wherein

a scaling factor is determined (606) from the present or expected relationship between the performance indicator and the operating target; and wherein

the scaling magnitude is calculated (608) from the scaling factor and a resource quantity presently allocated to the application.

4. The method of claim 2 or 3, wherein

the scaling magnitude is calculated taking into account multiple second performance measurement results obtained for multiple performance indicators, wherein for each performance indicator a dedicated operating target is defined, and wherein the scaling magnitude is calculated based on present or expected relationships between the performance indicators and the associated operating targets.

5. The method of any of the preceding claims, wherein
there exists a known correlation between the second performance measurement result and the resource quantity to be added to or removed from the application, and wherein the correlation is taken into account in the calculation of the scaling magnitude.

6. The method of claim 5 in combination with at least claim 2, wherein
the scaling magnitude is determined from the correlation and the relationship between the operating target and the performance indicator.

7. The method of claim 5 or 6, wherein
the correlation is a functional relationship or a mapping.

8. The method of any of claims 5 to 7, wherein
the correlation has been determined prior to runtime of the application.

9. The method of any of the preceding claims, wherein
the second performance measurement result is indicative of a system performance of the application having the set of virtual machines.

10. The method of any of claim 9, wherein
the second performance measurement result has been obtained by aggregating (70) individual performance measurements over the set of virtual machines.

11. The method of any of the preceding claims, wherein
at least one of the first measurement result and the second measurement result is indicative of a load of the application.

12. The method of any of the preceding claims, wherein
at least one of the first measurement result and the second measurement result is independent of the number of virtual machines associated with the application.

13. The method of any of the preceding claims, wherein
the first measurement result is used as the second measurement result in the calculation of the scaling magnitude.

14. The method of any of the preceding claims, further comprising

determining (302; 604) that a scaling operation is required by subjecting the first performance measurement result to at least one threshold decision.

15. The method of claim 14 in combination with at least claim 2, wherein

a lower threshold and an upper threshold are defined, and wherein the operating target lies between the lower threshold and the upper threshold.

16. The method of any of the preceding claims, further comprising

verifying the calculated scaling magnitude; and

adjusting the calculated scaling magnitude dependent on a result of the verification, wherein the scheduling request is triggered to be generated to be indicative of the adjusted scaling magnitude.

17. The method of claim 16, wherein

verifying the calculated scaling magnitude comprises comparing the calculated scaling magnitude with at least one configuration parameter.

18. The method of any of the preceding claims, wherein

the resource quantity is indicative of at least one of a number of virtual machines and an amount of infrastructure resources of the virtual machines.

19. The method of claims 17 and 18, wherein

the at least one configuration parameter is selected from the parameter set comprising: a maximum number of allowed virtual machines for the application; a minimum number of allowed virtual machines for the application; a maximum amount of allowed infrastructure resources for an individual virtual machine; a minimum amount of allowed infrastructure resources for an individual virtual machine.

20. A computer program product comprising program code portions for performing the steps of any of the preceding claims when the computer program product is executed on one or more computing devices.

21. The computer program product of claim 20, stored on a computer-readable recording medium.

22. An arrangement (20) configured to trigger scaling of an application (42) having a set of one or more virtual machines (46), the arrangement comprising at least one processor (22) configured to perform the following operations during runtime of the application and responsive to a determination (302; 604) that a scaling operation is required for the application, wherein the determination is based on at least one first performance measurement result obtained for the application:

calculate (304; 606, 608) a scaling magnitude for the required scaling operation taking into account at least one second performance measurement result obtained for the application, wherein the scaling magnitude is indicative of a resource quantity to be added to or removed from the application; and

trigger generation (306; 610) of a scaling request, wherein the scaling request is directed at a scaling of the application based on the calculated scaling magnitude.

23. The arrangement of claim 22, wherein

the arrangement is configured to perform the steps of any of claims 2 to 19.

24. A system (100) comprising the arrangement of any of claims 22 and 23 and the application having the set of one or more virtualised machines.

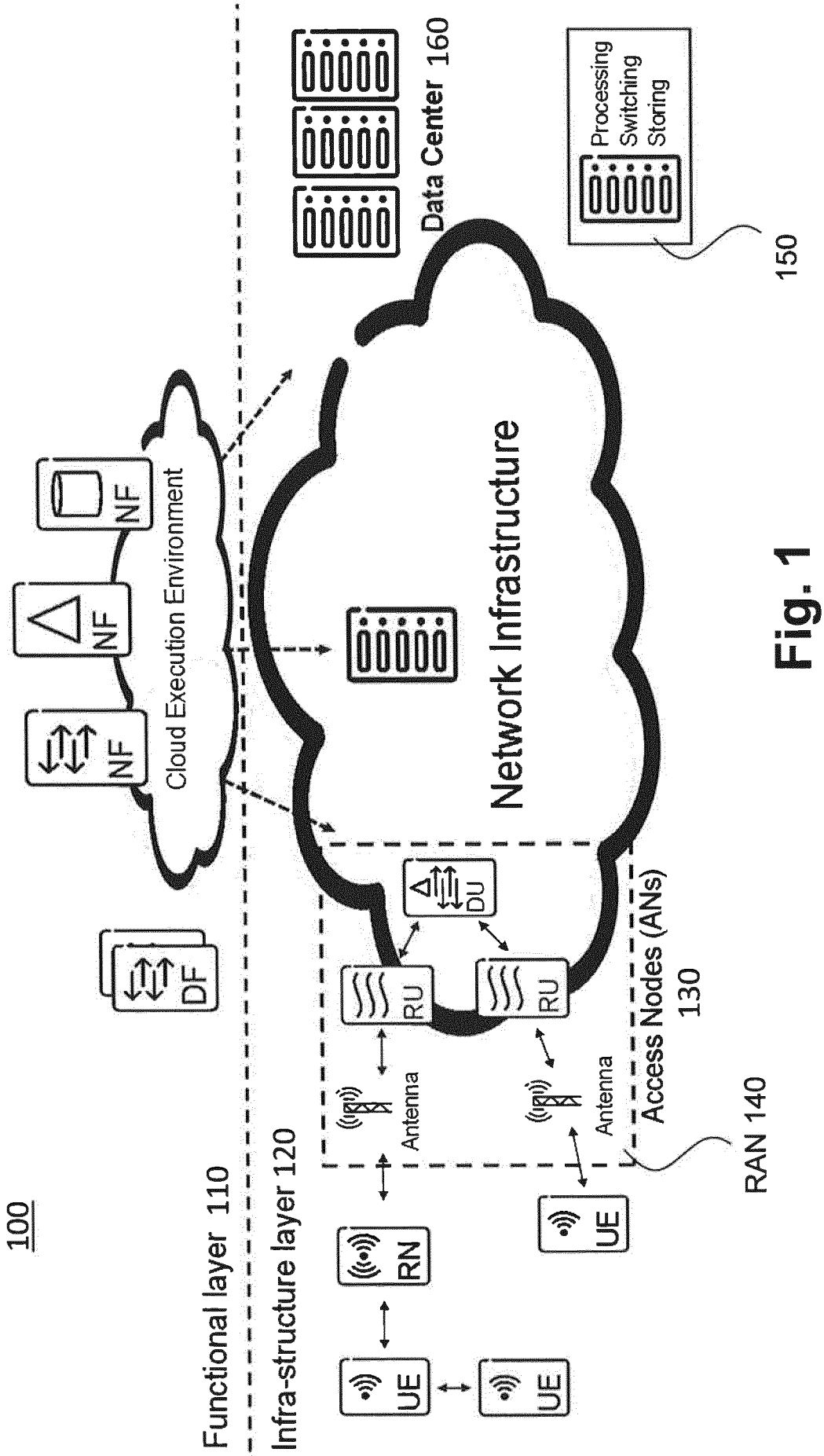
100
Functional layer 110
Infra-structure layer 120
Cloud Execution Environment
NF
NF
NF
DF
Network Infrastructure
Data Center 160
Processing
Switching
Storing
150
DU
RU
RU
Antenna
Antenna
Access Nodes (ANs)
130
RAN 140
RN
UE
UE
UE

Fig. 1

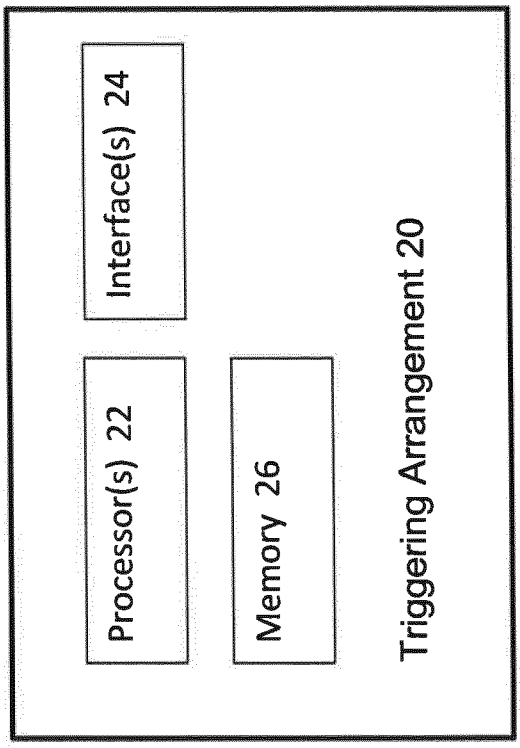
Processor(s) 22
Interface(s) 24
Memory 26
Triggering Arrangement 20

Fig. 2

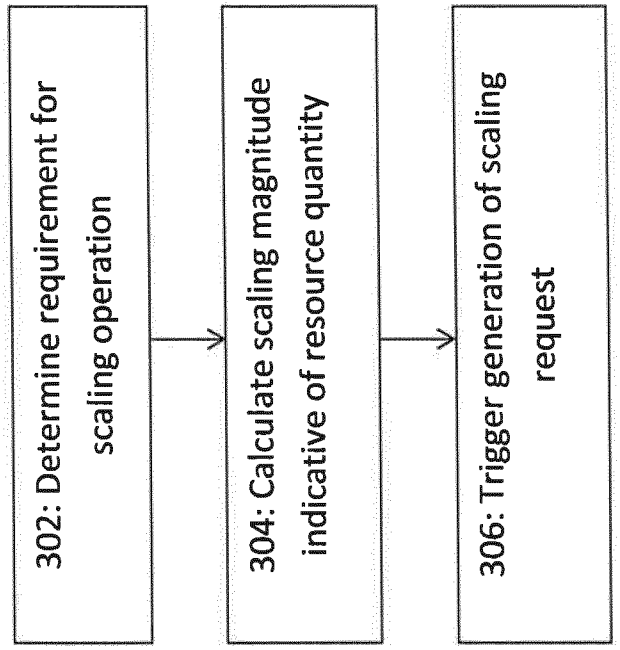
302: Determine requirement for scaling operation
304: Calculate scaling magnitude indicative of resource quantity
306: Trigger generation of scaling request

Fig. 3

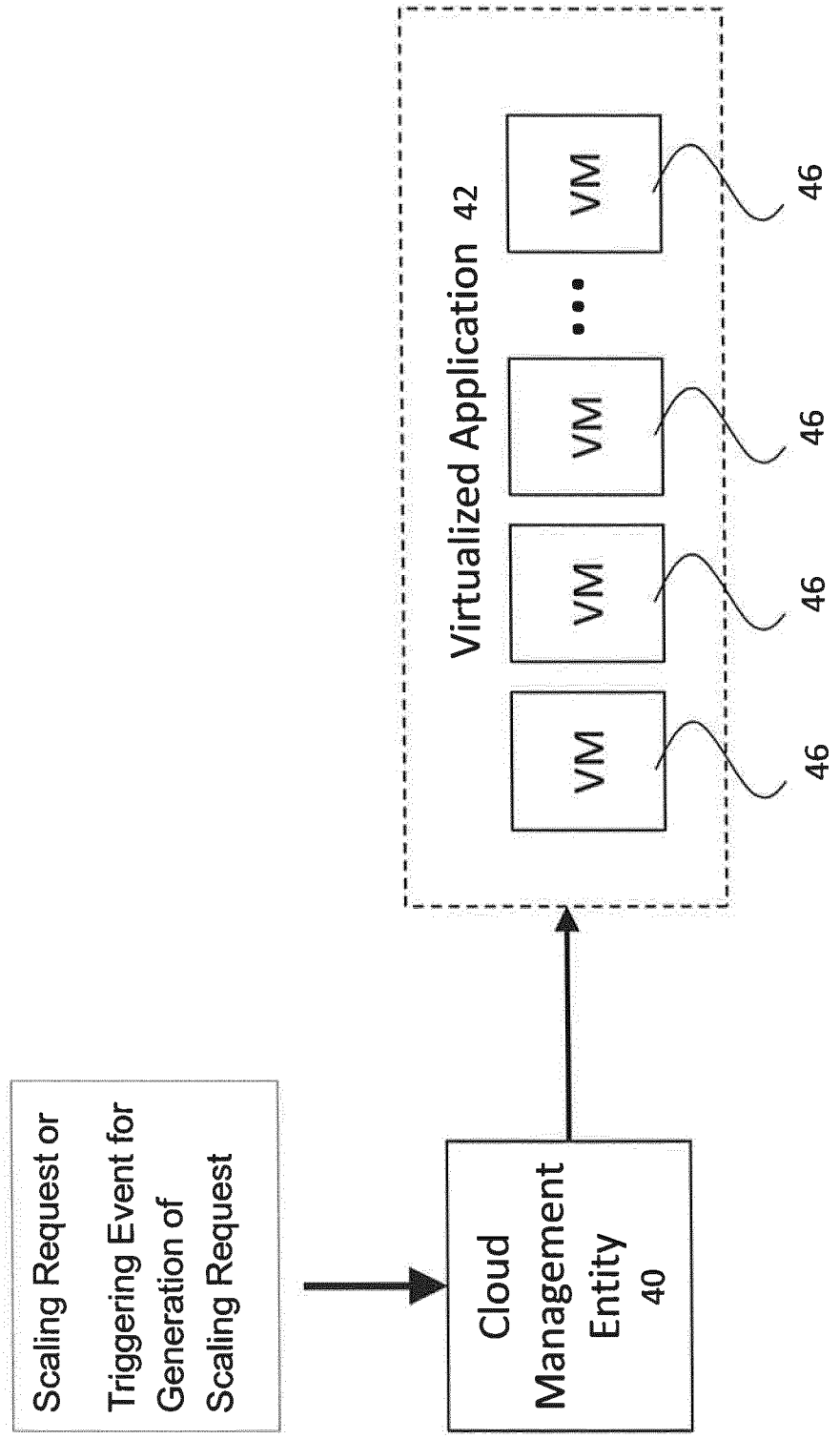
Scaling Request or
Triggering Event for
Generation of
Scaling Request
Cloud
Management
Entity
40
Virtualized Application 42
VM
VM
VM
VM
46
46
46
46

Fig. 4

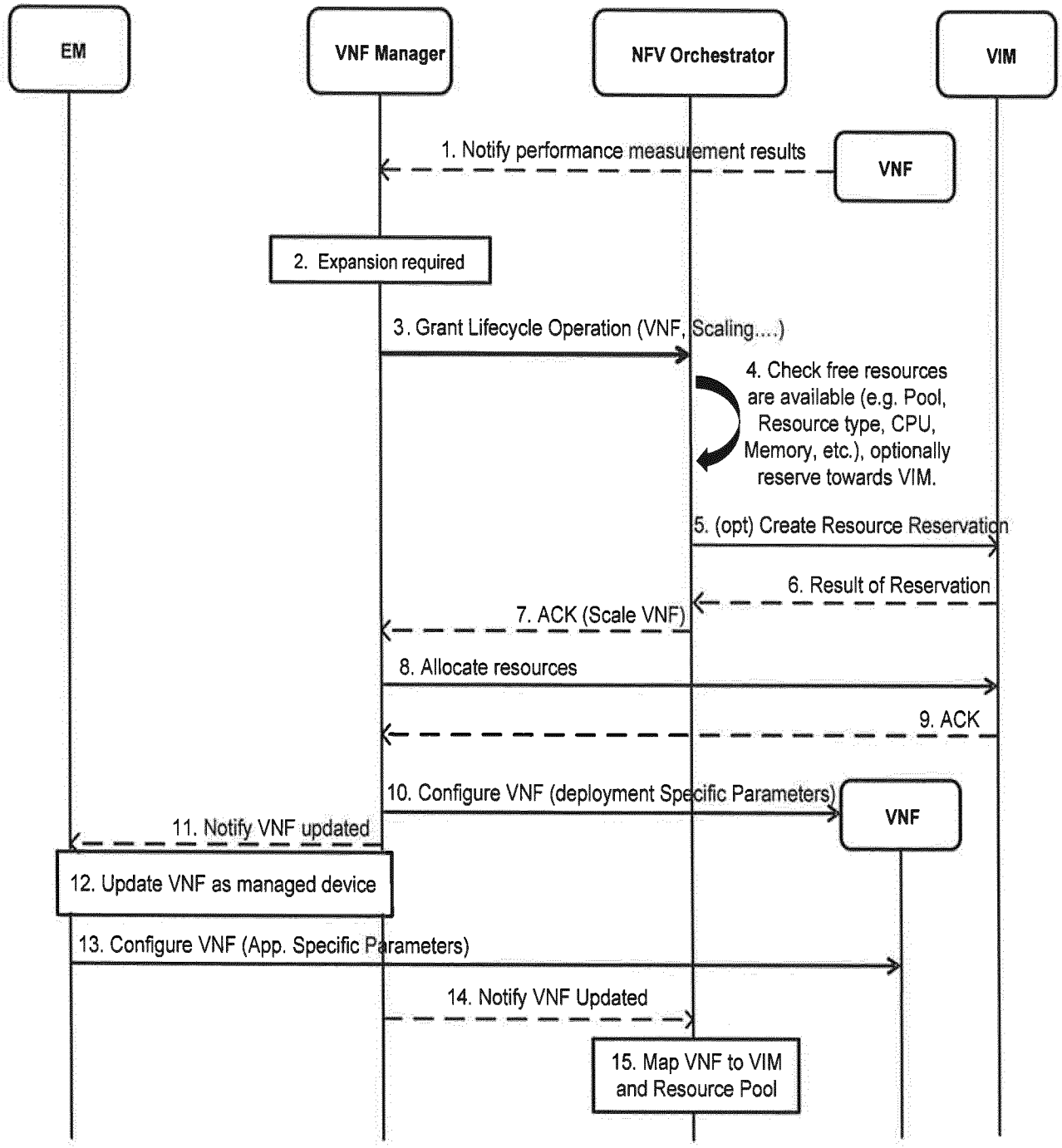
EM
VNF Manager
NFV Orchestrator
VIM
1. Notify performance measurement results
VNF
2. Expansion required
3. Grant Lifecycle Operation (VNF, Scaling....)
4. Check free resources are available (e.g. Pool, Resource type, CPU, Memory, etc.), optionally reserve towards VIM.
5. (opt) Create Resource Reservation
6. Result of Reservation
7. ACK (Scale VNF)
8. Allocate resources
9. ACK
10. Configure VNF (deployment Specific Parameters)
VNF
11. Notify VNF updated
12. Update VNF as managed device
13. Configure VNF (App. Specific Parameters)
14. Notify VNF Updated
15. Map VNF to VIM and Resource Pool

Fig. 5

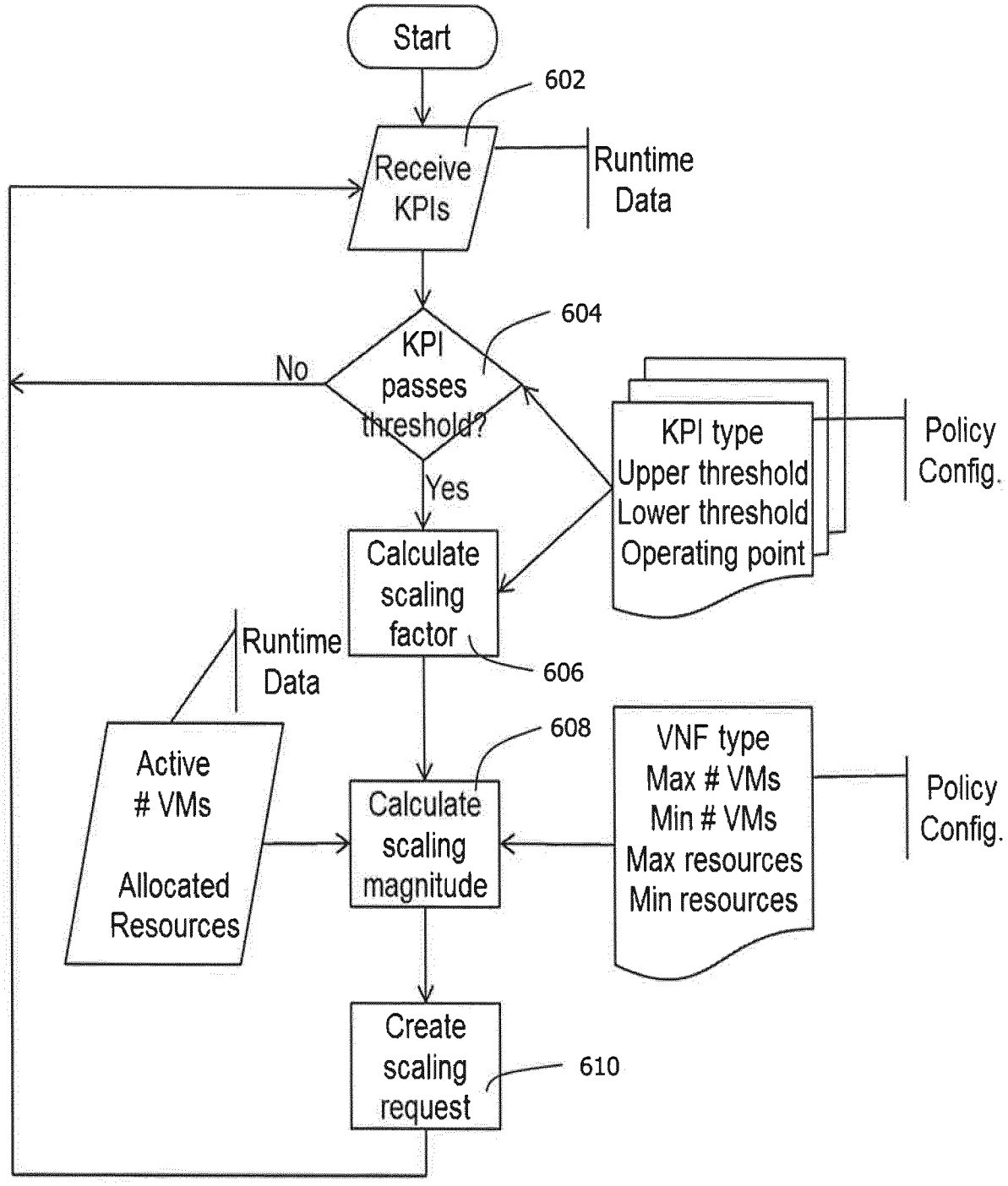
Start
602
Receive KPIs
Runtime Data
604
KPI passes threshold?
No
Yes
KPI type
Upper threshold
Lower threshold
Operating point
Policy Config.
Calculate scaling factor
606
Runtime Data
Active # VMs
Allocated Resources
608
Calculate scaling magnitude
VNF type
Max # VMs
Min # VMs
Max resources
Min resources
Policy Config.
Create scaling request
610

Fig. 6

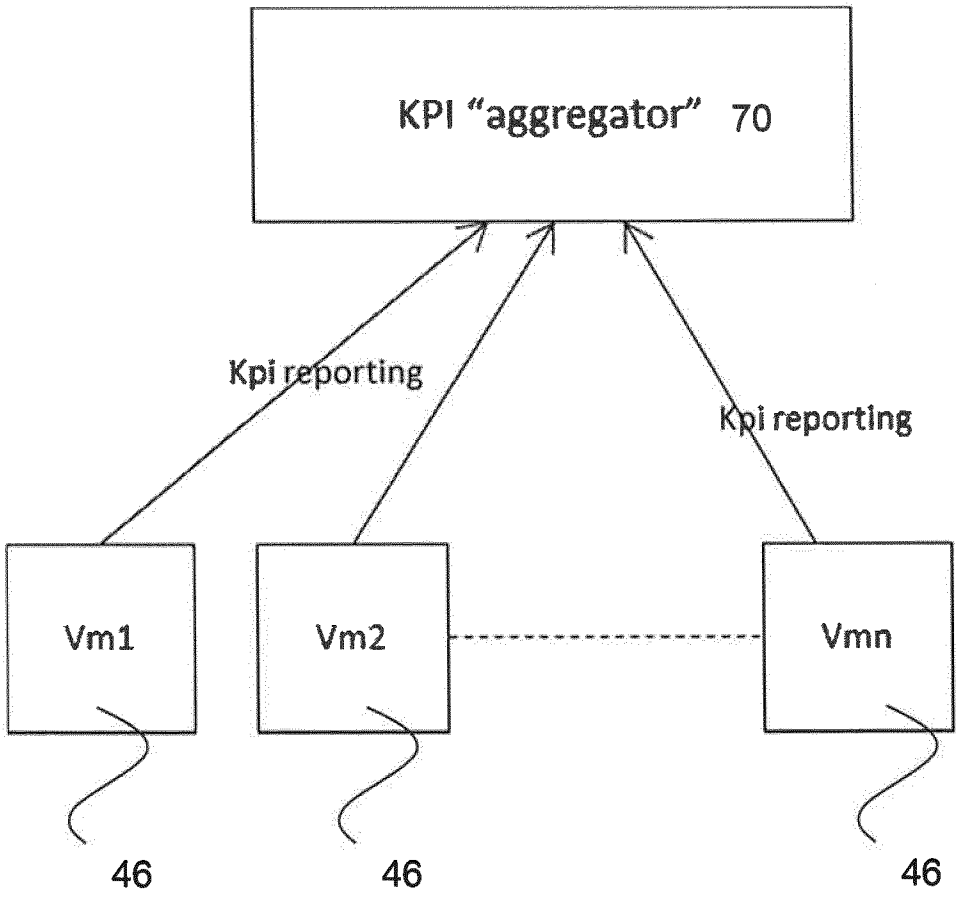
KPI "aggregator" 70
Kpi reporting
Kpi reporting
Vm1
Vm2
Vmn
46
46
46

Fig. 7

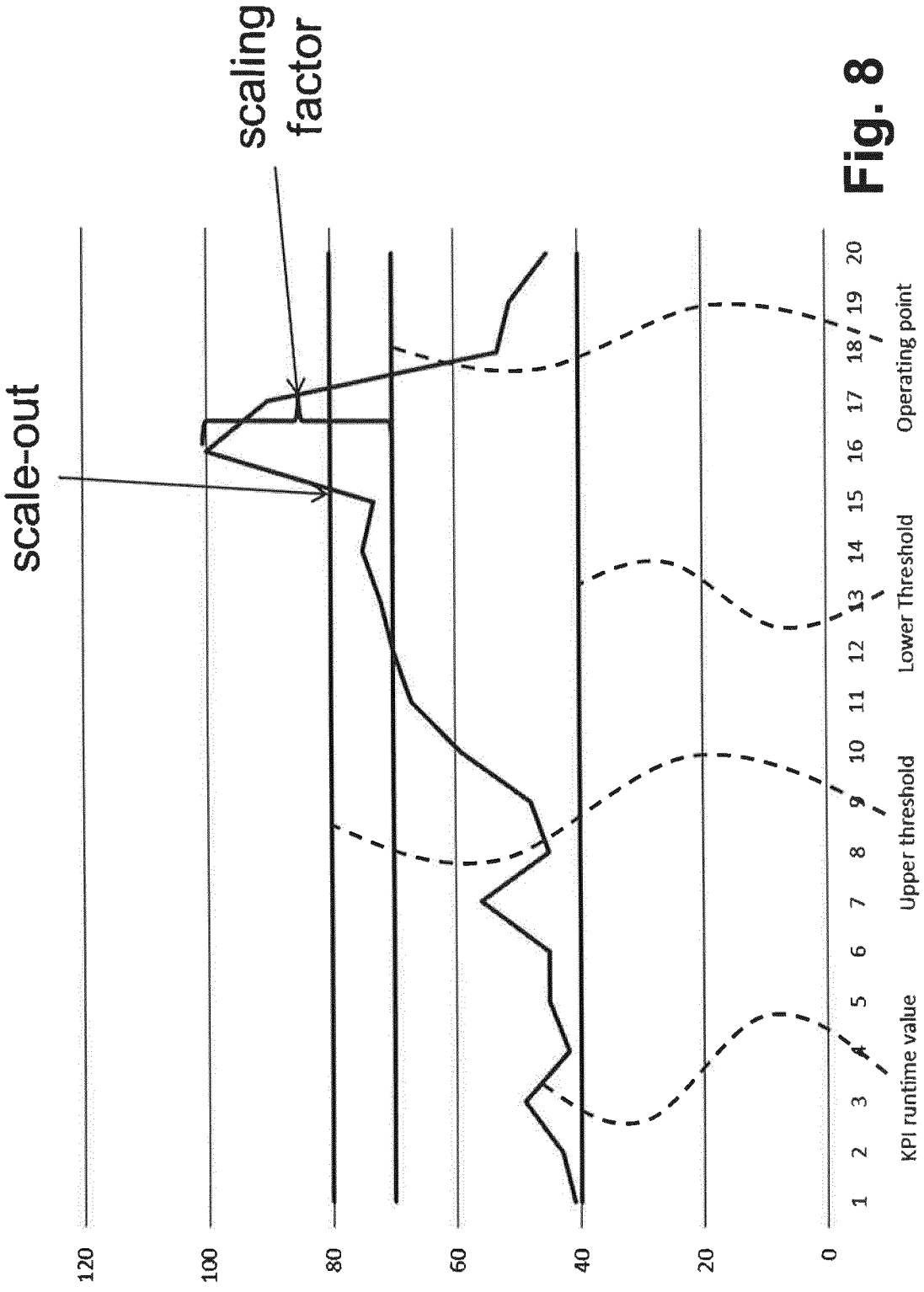
scaling factor
scale-out
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
KPI runtime value
Upper threshold
Lower Threshold
Operating point

Fig. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/057344

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/50
ADD.

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)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal , WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012/254443 A1 (UEDA YOHEI [JP]) 4 October 2012 (2012-10-04) abstract paragraph [0007] - paragraph [0008] paragraph [0034] - paragraph [0036] paragraph [0052] paragraph [0059] - paragraph [0082] -----	1-24
A	US 2014/040885 A1 (DONAHUE JAMES [US]) 6 February 2014 (2014-02-06) abstract paragraph [0031] - paragraph [0037] -----	1-24
A	US 2005/044228 A1 (BIRKESTRAND DANIEL C [US] ET AL) 24 February 2005 (2005-02-24) abstract paragraph [0093] - paragraph [0097] -----	1-24
	-/--	

[X] Further documents are listed in the continuation of Box C.

[X] 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" documentwhich may throw doubts on priority claim(s) orwhich 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	Date of mailing of the international search report
19 November 2015	30/11/2015

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Archontopoulos, E

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/057344

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 2 381 363 A2 (VMWARE INC [US]) 26 October 2011 (2011-10-26) abstract paragraph [0015]	1-24
A	US 2013/326506 Al (MCGRATH MICHAEL P [US] ET AL) 5 December 2013 (2013-12-05) abstract paragraph [0058] paragraph [0062] - paragraph [0068]	1-24
A	US 2011/179132 Al (MAYO MARK G [CA] ET AL) 21 July 2011 (2011-07-21) abstract paragraph [0024] - paragraph [0027] paragraph [0033]	1-24
A	US 2014/201374 Al (ASHW00D-SMITH PETER [CA] ET AL) 17 July 2014 (2014-07-17) abstract paragraph [0024] - paragraph [0025]	1-24

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2015/057344

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012254443 Al	04-10-2012	JP 2012208781 A	25-10-2012
		US 2012254443 Al	04-10-2012
US 2014040885 Al	06-02-2014	NONE	
US 2005044228 Al	24-02-2005	NONE	
EP 2381363 A2	26-10-2011	AU 2011201795 Al	10-11-2011
		EP 2381363 A2	26-10-2011
		JP 5318903 B2	16-10-2013
		JP 2011233146 A	17-11-2011
		US 2011265164 Al	27-10-2011
		US 2014130038 Al	08-05-2014
US 2013326506 Al	05-12-2013	US 2013326506 Al	05-12-2013
		US 2015081916 Al	19-03-2015
us 2011179132 Al	21-07-2011	US 2011179132 Al	21-07-2011
		US 2011179134 Al	21-07-2011
		US 2011179162 Al	21-07-2011
		Wo 2011088224 A2	21-07-2011
us 2014201374 Al	17-07-2014	EP 2936754 Al	28-10-2015
		KR 20150105421 A	16-09-2015
		US 2014201374 Al	17-07-2014
		Wo 2014110453 Al	17-07-2014