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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 November 2017 (16.11.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/194249 A1

(51) International Patent Classification:
G06F 9/445 (2006.01) *G06F 9/50* (2006.01)

(21) International Application Number:
PCT/EP2017/058383

(22) International Filing Date:
07 April 2017 (07.04.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
16290076.5 09 May 2016 (09.05.2016) EP

(71) Applicant: **ALCATEL LUCENT** [FR/FR]; 148/152 Route de la Reine, 92100 Boulogne-Billancourt (FR).

(72) Inventors: **CHEN, Ruichuan**; Alcatel-Lucent Deutschland AG, Lorenzstrasse 10, 70435 Stuttgart (DE). **VISWANATH, Bimal**; Alcatel-Lucent Deutschland AG, Lorenzstrasse 10, 70435 Stuttgart (DE). **AKKUS, Istemi Ekin**; Alcatel-Lucent Deutschland AG, Lorenzstrasse 10, 70435 Stuttgart (DE). **RIMAC, Ivica**; Alcatel-Lucent Deutschland AG, Lorenzstrasse 10, 70435 Stuttgart (DE). **HILT, Volker**; Alcatel-Lucent Deutschland AG, Lorenzstrasse 10, 70435 Stuttgart (DE).

(74) Agent: **WETZEL, Emmanuelle**; Alcatel-Lucent Deutschland AG, Lorenzstrasse 10, 70435 Stuttgart (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, DJ, 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, KH, KN, KP, KR, KW, 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.

WO 2017/194249 A1

(54) Title: METHOD OF AND DEVICE FOR DEPLOYING AN APPLICATION RELIABLY IN A COMPUTER NETWORK

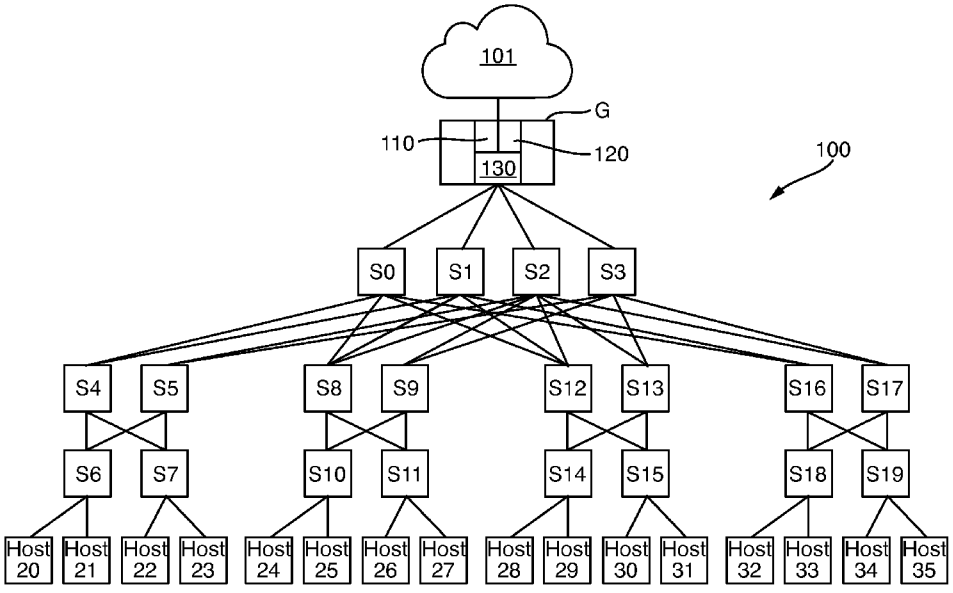

Fig. 1

(57) Abstract: The invention concerns a Method and corresponding device of deploying an application reliably in a computer network (100) comprising: Determining (204) information about a reliability of a predetermined first deployment for the application, Determining (207) information about a reliability of a predetermined second deployment for the application, Comparing (208) the information about the reliability of the first deployment with the information about the reliability of the second deployment, Selecting (208) the first deployment or the second deployment for deploying the application in the computer network (100) depending on the result of the comparison.

[Continued on next page]

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

Declarations under Rule 4.17:

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

— *of inventorship (Rule 4.17(iv))*

Published:

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

Title: Method Of And Device For Deploying An Application Reliably In A Computer Network

Specification

Field of the invention

The invention concerns a method of and a device for deploying an application reliably in a computer network, in particular a large scale data center.

Background

Applications are deployed in data center infrastructures to maintain high reliability.

WO 2012/105980 A1 discloses technologies for deploying an application in a ubiquitous computing system.

The paper Reliability-driven deployment optimization for embedded systems by Indika Meedeniya et. al. Journal of systems & software, Elsevier north Holland, New York, NY, US, vol. 84, no. 5, 3 January 2011, pages 835-849 discloses a reliability-driven deployment of components for embedded systems.

One technique to increase the reliability of these applications is to use redundancy. The developers of an application deploy multiple replicas of their application into large scale data centers. However, such redundancy efforts can be undermined if the reliability of underlying components does not meet the expectations. For example some or all replicas may fail at the same time due to the common hardware or software dependency shared by replicas.

This problem is increasingly relevant, as developers have limited control over the placement of their application in a data center operated by data center providers. Finding a reliable deployment for an application in particular in a large scale data center, has been proven to be NP-hard, meaning there is no polynomial-time solution.

Therefore it is desirable to provide mechanisms for deploying an application and for application developers to assess the reliability of an application deployment in a data center, e.g. using a part of an entire data center infrastructure.

Summary

This goal is achieved by a method and corresponding device. The method of deploying an application reliably in a computer network comprises:

Determining information about the reliability of a predetermined first deployment for the application,
Determining information about the reliability of a predetermined second deployment for the application,
Comparing the information about the reliability of the first deployment with the information about the reliability of the second deployment,
Selecting the first deployment or the second deployment for deploying the application in the computer network (100) depending on the result of the comparison.

This way the reliability of an application deployment in a computer network, e.g. of a data center or computer cloud is assessed and a deployment plan for an application is searched and found that meets the given reliability criteria.

Preferably the steps of determining, comparing and selecting are repeated until a predetermined number of repetitions is reached, a predetermined time period has passed, or a predetermined reliability threshold has been reached by the selected deployment.

Preferably the information about the reliability is determined by

Generating respective failure states for a plurality of devices for deploying the application according to a predetermined deployment,

Determining information about the reachability of the plurality of devices depending on the generated failure states,

Determining the information about the reliability depending on the information about the reachability.

Preferably the information about the reliability is determined for all devices for deploying the application according to the predetermined deployment.

Preferably a deployment with a higher reliability is selected over a deployment with a lower reliability.

Preferably a deployment with a lower reliability is selected over a deployment with a higher reliability, depending on a predetermined acceptance probability.

Preferably the reliability of a deployment is assessed by randomly generating a failure state for respective components of the computer network based on a predetermined failure probability of the respective component.

Preferably the method comprises receiving information about the failure probability and the respective component.

Preferably a deployment for assessing the reliability is selected from a plurality of deployments available in the computer network using a predetermined variance reduction criterion.

Preferably the computer network has one of a Fat Tree, BCube or DCell topology.

Preferably the computer network comprises at least one network node for accessing the computer network and a plurality of hosts for executing respective replica of the application according to a respective predetermined deployment, and wherein the information about the reliability of the application is determined depending on the connectivity between the at least one network node and respective hosts according to the predetermined deployment.

Preferably the information about the reliability is assessed depending on connectivity among different replicas of the application.

Further developments of the invention can be gathered from the dependent claims and the following description.

Brief description of the figures

Fig. 1 schematically depicts part of a computer network,

Fig. 2 schematically depicts steps of an exemplary method.

Description of the embodiments

Figure 1 schematically depicts part of a computer network 100. Components of the computer network are devices like

hosts for executing an application, network nodes, such as switches or gateways, and data links connecting them.

In the example computer network 100 is part of a data center. In the example in figure 1 the computer network 100 in the data center has a fat tree topology for connecting network nodes. For illustration purposes only, it is assumed that the data center consist of 16 hosts labelled in figure 1 as Host 20, …, Host 35.

However the invention is not limited to this exact network topology or number of network nodes. The invention also applies for different numbers of network nodes, data links or other topologies, e.g. BCube, DCell, and when multiple data centers or computer networks are connected.

The computer network 100 in the example comprises a gateway G. Gateway G is a network node that connects to the Internet, symbolized by a cloud 101. In the example, Ethernet or Transmission Control Protocol / Internet Protocol are used for connecting network nodes and Internet via data links.

Hosts in the example are network nodes adapted to execute a copy of an application. Copies of an application are referred to as replica of the application.

In the example any host may or may not be adapted to execute a replica of the application.

The application is accessible from the Internet on any of the hosts adapted to execute the application via gateway G. The application may be accessible from the Internet via different network nodes as well.

In computer network 100 gateway G is connected to core switches S0, S1, S2, S3 via respective data links.

Core switches S0 and S1 are connected to switches S4, S8, S12 and S16 with respective data links. Core switches S2 and S3 are connected to switches S5, S9, S13 and S17 with respective data links.

Switches S4 and S5 are connected to switches S6 and S7 with respective data links. Switch S6 is connected to Host 20 and Host 21 with respective data links. Switch S7 is connected to Host 22 and Host 23 with respective data links.

Switches S8 and S9 are connected to Switches S10 and S11 with respective data links. Switch S10 is connected to Host 24 and Host 25 with respective data links. Switch S11 is connected to Host 26 and Host 27 with respective data links.

Switches S12 and S14 are connected to switches S14 and S15 with respective data links. Switch S14 is connected to Host 28 and Host 29 with respective data links. Switch S15 is connected to Host 30 and Host 31 with respective data links.

Switches S16 and S17 are connected to switches S18 and S19 with respective data links. Switch S18 is connected to Host 32 and Host 33 with respective data links. Switch S19 is connected to Host 34 and Host 35 with respective data links.

The aforementioned switches are network nodes in the computer network 100. Other network nodes, e.g. routers or further gateways may be used as well.

A method to assess the reliability of any specific application deployment in a data center, and to efficiently find the reliable deployment for an application even in a large-scale data center is described below referencing Figure 2.

In the method the reliability of application deployments is assessed by executing many simulation trials.

In each trial, certain components are considered as failed according to their failure probabilities.

The output of each trial is the reliability score of a specific deployment plan for the application, i.e., how likely K out of N replicas used in the deployment are reachable in this trial.

Eventually, a reliability score of each trial is taken into account, and a search for the most reliable deployment plan for this application is conducted.

Considering a deployment where the application runs with N replicas of the application, preferably on N different hosts, the application is considered to be reliable as long as K out of the N replicas are reachable from the gateway G, which is in the example connected to the Internet. This means that computer network 100 comprises at least one network node, e.g. gateway G, for accessing the computer network 100 and a plurality of devices, e.g. hosts Host 20, …, Host 35, for executing respective replica of the application according to a respective predetermined deployment. In this example the information about the reliability of the application is determined depending on

the connectivity between the at least one network node and respective hosts according to the predetermined deployment.

Alternatively the information about the reliability may be assessed depending on connectivity among different replica of the application. This way, for example modelling of a chain of service within the application is supported by using one of the replicas of the application as the destination for example instead of the gateway G.

To assess a deployment some of the components in the topology of computer network 100 are intentionally set to a failed state. Then, for example, the Fat-Tree routing protocol is run to check how many replicas of the application can be reached by the gateway G.

This fail-and-check process is repeated many rounds, e.g. for Y rounds. Eventually, if at least K application replicas are reachable in X rounds, the reliability of the corresponding deployment is considered to be X/Y.

A component, preferably each component, fails in the example with a specific probability. It is assumed that an operator of the data center is running a management system for managing the data center components and/or infrastructure. The management system in the example monitors components and their respective failure probabilities.

The management system in the example provides an interface adapted to send information about a failure probability and a corresponding component of the computer network.

A corresponding device, in the example gateway G, for deploying an application in computer network 100 comprises

a processor 110 and an interface 120 to the managing system. The interface is preferably adapted to receive information about the failure probability and the corresponding component of the computer network 100. The processor 110 is preferably adapted to select a deployment for the application according to the method described below.

Preferably the device, e.g. gateway G, is configured with an output 130 to output the selected deployment or to deploy the application in the computer network 100 according to the selected deployment by output of instructions for configuring respective network nodes of the computer network according to the selected deployment. Output 130 is for example a data interface for a terminal for an application developer or to send the instructions. The information provided by output 130 to the application developer is providing internal machine states of the computer network, that guide the application developer to the deployment required for the application to run with best possible or a predetermined reliability. The predetermined reliability target may be input to the processor 110 by interface 120 as a parameter.

According to the method, after the start, in a step 201 the information about the failure probability and corresponding components is received.

Afterwards in a step 202, a deployment for assessing the reliability is selected from a plurality of deployments available in the computer network. The plurality of deployments available in the computer network may be provided by the management system via the interface, or generated from the information regarding the components received in step 201. Preferably the reliability of each

deployment is determined using a predetermined variance reduction criterion.

In an initialization, i.e. at first execution of step 202, a first deployment is selected as an initial deployment. In the example the initial deployment is determined by assigning random hosts to a random number of replicas of the application.

Afterwards a step 203 is executed. In step 203 the reachability of the first deployment is assessed. In the example the assessment is based on Monte-Carlo sampling, i.e. randomly generated failure states are assumed for each component according to the component's failure probability.

To this end, the generation of failure states is repeated to randomly generate the failure states for each component across the predetermined number of rounds Y. This produces for example the following table:

	S0	**S1**	…	**S19**	**Host 20**	**Host 21**	…	**Host 35**
Round 1	Alive	Alive	…	Failed	Alive	Failed	…	Alive
Round 2	Failed	Failed	…	Alive	Alive	Alive	…	Alive
……	……	……	…	……	……	……	…	……
Round Y	Alive	Failed	…	Failed	Failed	Alive	…	Alive

In the table above each row represents one round of sampling, and each column represents the failure states of the corresponding component across all the rounds.

The above-mentioned Monte-Carlo-sampling-based failure state generation provides a very simple assignment of the failure states to the corresponding components.

The number of rounds Y may be selected depending on the required accuracy because the generation of failure states can be repeated for multiple rounds to increase accuracy. Preferably the information about the reachability is determined for all devices for deploying the application according to the predetermined deployment. Large-scale data centers may have at least tens of thousands of components. The components in data centers normally fail with low probabilities, e.g., 10^{-3} or even lower.

Considering all devices may affect the speed of processing, to speed up the failure state generation, a variance reduction technique may be used. The speed of failure state generation is significantly increased by using for example Dagger Sampling. This method and other methods for variance reduction are specifically designed to enable efficient Monte-Carlo sampling for rare events in large-scale networks. Performing corresponding variance reduction is referred to as using a predetermined variance reduction criterion.

This way information about the reachability is determined by generating respective failure states for a plurality of hosts for deploying the application according to a predetermined deployment. Furthermore information about the reachability of the plurality of hosts is determined this way depending on the generated failure states.

Afterwards a step 204 is executed. In step 204, information about the reliability is determined for hosts for deploying the application according to the first deployment depending on the information about the reachability determined in step 203.

Preferably the randomly generated failure states for all components in each round are used. Then, for a host on which an application replica is being deployed, the Fat-Tree routing protocol is run to check whether this deployed host is reachable from the gateway G. This check is repeated for all the N hosts which are being used to deploy the application replicas.

If at least K out of these N hosts are reachable in this round, then this deployment is considered reliable for this round.

	Round 1	**Round 2**	**……**	**Round Y**
At least K of N deployed hosts reachable	Yes	No		Yes

This is repeated for all the Y rounds. An exemplary result is given in the table above.

Preferably if the deployment is reliable in X rounds, then the reliability of this deployment is determined as X/Y.

In the example X is a number of rounds that is smaller than the predetermined number of rounds Y.

This way information about the reliability of the predetermined first deployment for the application is determined.

Afterwards a step 205 is executed. In step 205, a second deployment for assessing the reliability is selected from the plurality of deployments available in the computer network. In the example the second deployment is determined by replacing one or more of the hosts used in the first deployment by random new hosts. Likewise random hosts may

be assigned to another random number of replicas of the application as well.

Afterwards a step 206 is executed. In step 206, the reachability of the second deployment is assessed. In the example the assessment is made as described in step 203 above based on Monte-Carlo sampling. This means that randomly generated failure states are assumed for each component according to the component's failure probability

To this end, the generation of failure states is repeated to randomly generate the failure states for each component across a predetermined number of rounds. This produces for example a similar table as described in step 203.

This way information about the reliability of the predetermined second deployment for the application is determined.

Afterwards, a step 207 is executed. In step 207, the information about the reliability of the second deployment is determined for hosts for deploying the application according to the second deployment depending on the information about the reachability determined in step 206.

The information about the reliability is for example determined as described in step 204.

Afterwards a step 208 is executed.

In step 208 the information about the reliability of the first deployment is compared with the information about the reliability of the second deployment to select the first deployment or the second deployment for deploying the

application in the computer network depending on the result of the comparison.

Preferably a deployment with a higher reliability is selected over a deployment with a lower reliability.

Alternatively or additionally a deployment with a lower reliability may be selected over a deployment with a higher reliability, depending on a predetermined acceptance probability.

The steps 202 to 208 are preferably repeated for further comparison of the deployment selected in step 208 with other deployments. To that end for example after the initial deployment has been determined in step 202, instead of determining the first deployment in step 202 again, the deployment selected in step 208 is used for any further repetition of step 202.

Preferably the steps 202 to 208 are repeated until a predetermined number of trials has been reached, or a pre-defined time period has passed, or a required reliability has been reached by the current selected deployment. A corresponding Test is executed in a step 209 in the example between steps 208 and 202.

An example for different results of repetitions is as follows:

Trial	Hosts for deployment	Relia-bility	Acceptance Probability	Select
1	Host 20, Host 30	0.8	-	Yes
2	Host 23, Host 30	0.95	-	Yes
3	Host 23, Host 33	0.9	No	No
4	Host 25, Host 30	0.85	Yes	Yes
5	Host 25, Host 31	0.93	-	Yes
.....	.....	.....		.....

According to this example, in Trial 1, e.g. the first deployment is initially accepted as initial deployment. In Trial 2, e.g. the second deployment is selected as deployment with higher reliability.

In Trial 3, e.g. a third deployment is not selected, because the reliability is less than the reliability of the previously selected deployment. In this case an optionally used predetermined acceptance probability is not met.

In Trial 4, e.g. a fourth deployment is selected despite having less reliability than the previously selected deployment. In this case the optionally used predetermined acceptance probability is met.

Using the acceptance probability avoids converging to a local optimum.

In Trial 5, e.g. a fifth deployment is selected as deployment with higher reliability than the previously selected deployment.

Preferably among all the trials, the deployment with the highest reliability is reported as the best deployment. The report is for example an output of the selected deployment.

Alternatively or additionally the deployment of the application in the computer network is an output of

instructions for configuring respective network nodes of the computer network according to the selected deployment.

The description and drawings merely illustrate the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise various arrangements that, although not explicitly described or shown herein, embody the principles of the invention. Furthermore, all examples recited herein are principally intended expressly to be only for pedagogical purposes to aid the reader in understanding the principles of the invention and the concepts contributed by the inventor(s) to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the invention, as well as specific examples thereof, are intended to encompass equivalents thereof.

The functions of the various elements shown in the figures, including any functional blocks, may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Moreover, explicit use of the term 'processor' or 'controller' should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital signal processor (DSP) hardware, network processor, application specific integrated circuit (ASIC), field programmable gate array (FPGA), read only memory (ROM) for storing software, random access memory

(RAM), and non-volatile storage. Other hardware, conventional and/or custom, may also be included.

It should be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative circuitry embodying the principles of the invention. Similarly, it will be appreciated that any sequence diagrams represent various processes which may be substantially represented in computer readable medium and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

A person of skill in the art would readily recognize that steps of various above-described methods can be performed by programmed computers. Herein, some embodiments are also intended to cover program storage devices, e.g., digital data storage media, which are machine or computer readable and encode machine-executable or computer-executable programs of instructions, wherein said instructions perform some or all of the steps of said above-described methods. The program storage devices may be, e.g., digital memories, magnetic storage media such as a magnetic disks and magnetic tapes, hard drives, or optically readable digital data storage media. The embodiments are also intended to cover computers programmed to perform said steps of the above-described methods.

Claims

1. A computer implemented method of deploying an application in a computer network (100) comprising:

Determining (202, 203, 204) information about a reliability of a predetermined first deployment for the application,

Determining (205, 206, 207) information about a reliability of a predetermined second deployment for the application,

Comparing (208) the information about the reliability of the first deployment with the information about the reliability of the second deployment,

Selecting (208) the first deployment or the second deployment for deploying the application in the computer network (100) depending on the result of the comparison.

2. The method of claim 1, wherein the steps of determining (202,…, 207), comparing (208) and selecting (208) are repeated for further comparison of a deployment selected with other deployments, until a predetermined number of repetitions is reached, a predetermined time period has passed, or a predetermined reliability threshold has been reached by the current selected deployment.

3. The method of any of claims 1 or 2, wherein the information about the reliability is determined by

Generating (203, 206) respective failure states for a plurality of devices for deploying the application according to a predetermined deployment,

Determining (203, 206) information about the reachability of the plurality of devices depending on the generated failure states,

Determining (204, 209) the information about the reliability depending on the information about the reachability.

4. The method of claim 3, wherein the information about the reliability is determined for all devices for deploying the application according to the predetermined deployment.

5. The method of any of the previous claims, wherein a deployment with a higher reliability is selected over a deployment with a lower reliability.

6. The method of any of the claims 1 to 4, wherein a deployment with a lower reliability is selected over a deployment with a higher reliability, depending on a predetermined acceptance probability.

7. The method of any of the previous claims, wherein the reliability of a deployment is assessed by randomly generating a failure state for respective components of the computer network (100) based on a predetermined failure probability of the respective component.

8. The method of claim 7, comprising receiving (201) information about the failure probability and the respective component.

9. The method of any of claims 7 or 8, wherein a deployment for assessing the reliability is selected from a plurality of deployments available in the computer network (100) using a predetermined variance reduction criterion.

10. The method of any of the previous claims, wherein the computer network (100) has one of a Fat Tree, BCube or DCell topology.

11. The method of any of the previous claims, wherein the computer network (100) comprises at least one network node (G) for accessing the computer network (100) and a plurality of hosts (Host 20, … Host 35) for executing respective replica of the application according to a respective predetermined deployment, and wherein the information about the reliability of the application is determined depending on the connectivity between the at least one network node (G) and respective hosts (Host 20, … Host 35) according to the predetermined deployment.

12. The method of any of the previous claims, wherein the information about the reliability is assessed depending on connectivity among different replicas of the application.

13. A device (G) for deploying an application in a computer network (100) comprising a processor (110) adapted to:

Determine information about reliability of a predetermined first deployment for the application,

Determine information about reliability of a predetermined second deployment for the application,

Compare the information about the reliability of the first deployment with the information about the reliability of the second deployment,

Select the first deployment or the second deployment for deploying the application in the computer network (100) depending on the result of the comparison.

14. The device (G) according to claim 13, comprising an interface (120) for a management system of the computer network (100), adapted to receive information about a failure probability and a corresponding component of the computer network (100), and an output (130) for the selected deployment.

15. The device (G) according to claim 13 or 14, configured to deploy the application in the computer network (100) according to the selected deployment by output of instructions for configuring respective network nodes of the computer network (100) according to the selected deployment.

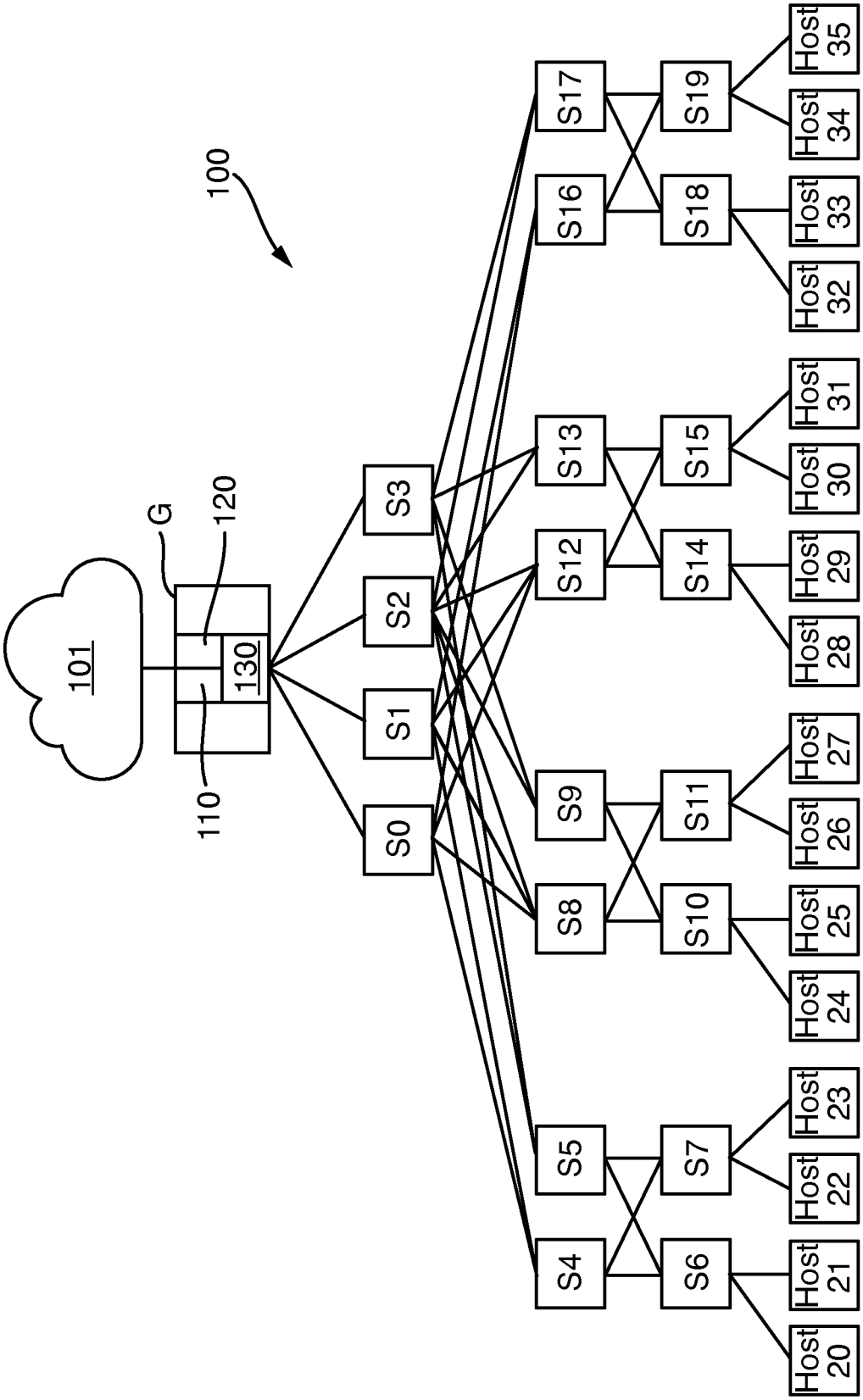
100
101
G
120
130
110
S0
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
Host 20
Host 21
Host 22
Host 23
Host 24
Host 25
Host 26
Host 27
Host 28
Host 29
Host 30
Host 31
Host 32
Host 33
Host 34
Host 35

Fig. 1

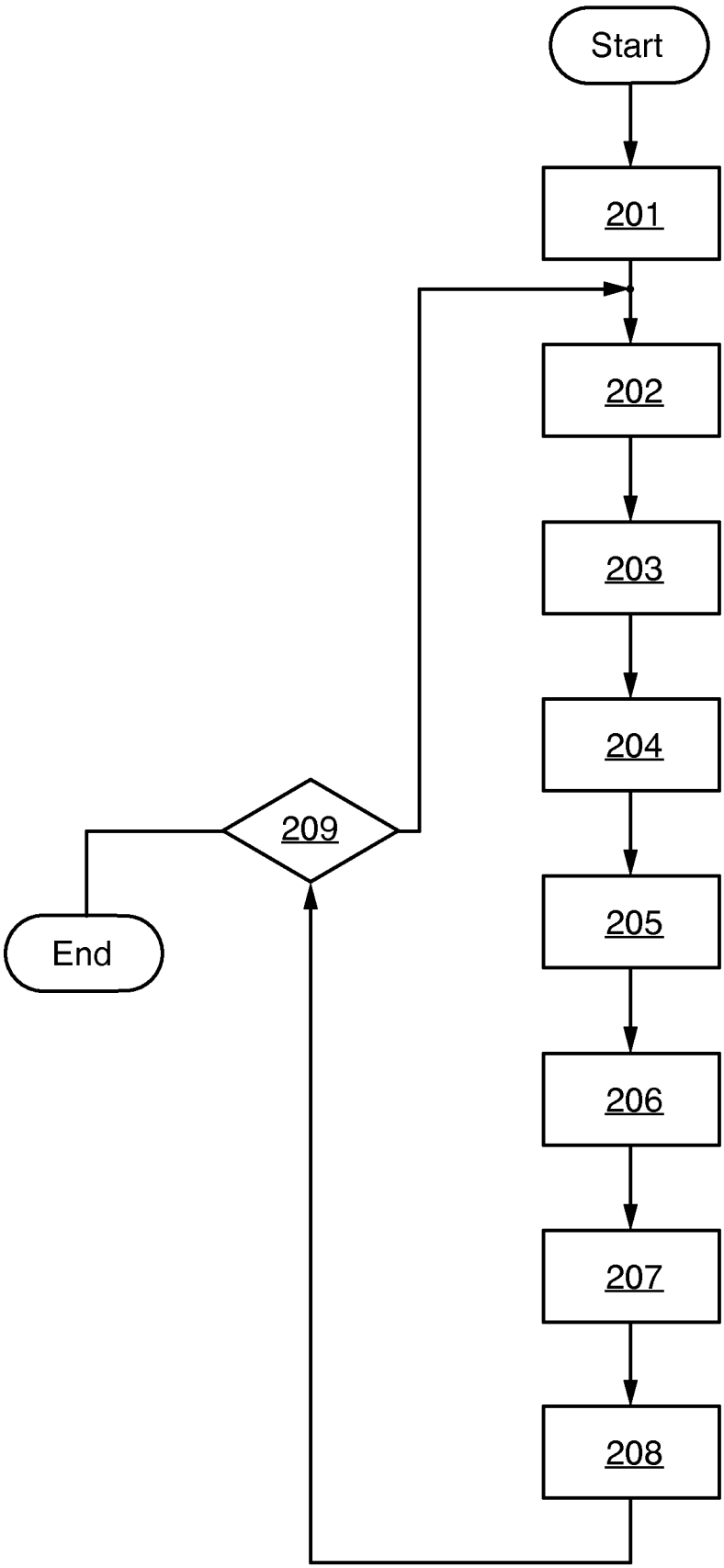

Fig. 2

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2017/058383

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/445 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, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2012/105980 A1 (EMPIRE TECHNOLOGY DEV LLC [US]; KRUGLICK EZEKIEL [US]) 9 August 2012 (2012-08-09) abstract page 5, paragraph 0013 - paragraph 0014 page 14, paragraph 0041 - paragraph 0042 page 8, paragraph 0024 - page 9 page 6, paragraph 0018 - page 7 ----- -/--	1-15

[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" 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	Date of mailing of the international search report
23 May 2017	07/06/2017

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	Hoareau, Samuel

1

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	INDIKA MEEDENIYA ET AL: "Reliability-driven deployment optimization for embedded systems", JOURNAL OF SYSTEMS & SOFTWARE, ELSEVIER NORTH HOLLAND, NEW YORK, NY, US, vol. 84, no. 5, 3 January 2011 (2011-01-03), pages 835-846, XP028170583, ISSN: 0164-1212, DOI: 10.1016/J.JSS.2011.01.004 [retrieved on 2011-01-28] abstract page 840, paragraph 4.2 -----	1-15
A	RAGHAVACHARI M ET AL: "The deployer's problem: configuring application servers for performance and reliability", SOFTWARE ENGINEERING, 2008. ICSE '08. ACM/IEEE 30TH INTERNATIONAL CONFERENCE ON; [INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING], IEEE, PISCATAWAY, NJ, USA, vol. CONF. 25, 3 May 2003 (2003-05-03), pages 484-489, XP010640144, ISSN: 0270-5257, DOI: 10.1109/ICSE.2003.1201226 ISBN: 978-0-7695-1877-0 abstract page 486, paragraph 3.1 -----	1-15
A	US 2014/201218 A1 (CATALANO JOHN [US] ET AL) 17 July 2014 (2014-07-17) abstract page 1, paragraph 0005 - page 2, paragraph 0011 page 4, paragraph 0038 - page 5, paragraph 0046 -----	1-15

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2017/058383

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2012105980 A1	09-08-2012	KR 20130101548 A	13-09-2013
		US 2013311628 A1	21-11-2013
		WO 2012105980 A1	09-08-2012
US 2014201218 A1	17-07-2014	NONE	