Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 2 April 2024

Case Number: T 1102/20 - 3.5.06

Application Number: 14811722.9

Publication Number: 3008616

IPC: G06F9/455, G06F9/50

Language of the proceedings: EN

Title of invention:
PRE-CONFIGURE AND PRE-LAUNCH COMPUTE RESOURCES

Applicant:
Amazon Technologies, Inc.

Headword:
Pre-configure and pre-launch compute resources/AMAZON

Relevant legal provisions:
EPC Art. 56

Keyword:
Inventive step - (no)

Decisions cited:

This datasheet is not part of the Decisior
It can be changed at any time and without notic

Catchword:

This datasheet is not part of the Decisior
It can be changed at any time and without notic

Europäisches Patentamt
European Patent Office
Office européen des brevets

Beschwerdekammern

Boards of Appeal

Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 1102/20 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 2 April 2024

Appellant: (Applicant)	Amazon Technologies, Inc. P.O. Box 81226 Seattle, WA 98108-1226 (US)
Representative:	Carpmaels & Ransford LLP One Southampton Row London WC1B 5HA (GB)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 25 November 2019 refusing European patent application No. 14811722.9 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	M. Müller
Members:	A. Teale
	B. Müller

Summary of Facts and Submissions

I. This is an appeal against the decision, dispatched with reasons on 25 November 2019, to refuse European patent application No. 14 811 722.9 on the basis that the subject-matter of claim 1 according to a main and an auxiliary request did not involve an inventive step, Article 56 EPC, in view of the following document:

D1: Shohei Yamasaki et al., "Model-based resource selection for efficient virtual cluster deployment", Virtualization Technology in Distributed Computing, ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA, 12 November 2007, pages 1 to 7, XP058164989, DOI: 10.1145/1408654.1408660, ISBN: 978-1-59593-897-8.

II. A notice of appeal against the decision in its entirety and the appeal fee were received on 5 February 2020.

III. The regular time limit for submitting the grounds of appeal expired on 5 April 2020. The statement of grounds of appeal was however received after this on 17 April 2020, this being the last day of the period announced in the Notice from the EPO dated 15 March 2020 (OJ EPO 2020, A29) concerning the disruption due to the COVID-19 outbreak for meeting time limits expiring on or after 15 March 2020. Hence the board considers the statement of grounds of appeal to be deemed received in due time. The appellant requested that a patent be granted on the basis of the main or the auxiliary request treated in the decision.

The appellant refiled said requests and made an auxiliary request for oral proceedings.

IV. In an annex to a summons to oral proceedings the board set out its provisional opinion on the appeal according to which it had doubts that the subject-matter of claims 1 and 3 of both requests involved an inventive step over D1, but also regarding the clarity, Article 84 EPC, of claims 1, 3 and 6 of both requests.

V. The appellant did not respond in substance to the board's preliminary opinion. However, in a letter, received on 28 February 2024, the appellant's representative stated that the appellant would not be represented at the scheduled oral proceedings. Thereupon the board cancelled the oral proceedings.

VI. The application is being considered in the following form:

Description (both requests):
pages 1 to 33, as published in WO 2014/201053 A1.

Claims (both requests received on 3 October 2019):
Main request: 1 to 6.
Auxiliary request: 1 to 6.

Drawings (both requests):
Pages 1/9 to 9/9, as published in WO 2014/201053 A1.

VII. Claim 1 according to the main request reads as follows:

"A system for provisioning and launching compute instances, the system comprising: one or more datacenters comprising a plurality of computing devices configured to communicate with each other over network

connections; and one or more memories having stored therein computer-readable instructions that, upon execution on the system, cause the system at least to: determine an expected demand for a plurality of compute instances, the plurality of compute instances being associated with a plurality of machine images and a plurality of compute instance types, wherein the expected demand comprises the number and types of compute instances that are expected to be requested; determine one or more computing devices from the plurality of computing devices configured to host the plurality of compute instances based at least in part on the expected demand, wherein the instructions to determine an expected demand for a plurality of compute instances comprise instructions that, upon execution on the system, cause the system at least to: analyze a history of requests for compute instances to determine a target for the plurality of compute instances and the one or more computing devices; and revise the target based at least in part on statistical factors associated with the requests; prior to receiving a request from a customer node to launch a compute instance, launch and idle the plurality of compute instances on the one or more computing devices based at least in part on the plurality of machine images and the plurality of compute instance types; and in response to receiving the request from the customer node to launch the compute instance, activate the compute instance from one of the plurality of machine images that are idle on the one or more computing devices; wherein the instructions to determine one or more computing devices from the plurality of computing devices comprise instructions that, upon execution on the system, cause the system at least to: determine an expected availability of the plurality of computing devices; balance the expected demand with the expected

availability to meet a constraint associated with launching and idling the plurality of compute instances; and determine the one or more computing devices based at least in part on the balance."

VIII. Claim 1 of the auxiliary request differs from that of the main request in the addition of the following passage:

"wherein the machine image is pre-configured for launch on the one or more computing devices based at least in part on: caching the machine image from a root storage device that stores the machine image; modifying a configuration file of the machine image based at least in part on a configuration of the compute instance; and allocating a storage volume to the compute instance based at least in part on the configuration of the compute instance".

Reasons for the Decision

1. Admissibility of the appeal

In view of the facts set out at points I to III above, the appeal fulfills the admissibility requirements under the EPC and is consequently admissible.

2. A summary of the invention

2.1 The application relates to cloud computing, in particular (see figure 1 and [19-27]) to responding more quickly to a request from a customer computing system (104) to provide compute instances (see figure 2; 206A-N and [29-38]), such as virtual machines (see

[3,30]), in one or more data centres (102A-N) accessed via a network (106), such as the internet; see [22].

2.2 This is achieved by gauging the expected demand (see figure 5) for compute devices and launching and idling compute devices (also referred to as "pre-warming"; see figure 6 and 7) in anticipation of a corresponding customer request (see figures 3 and 4 and [39-56]), which is then fulfilled more quickly using "pre-warmed" compute instances.

2.3 According to paragraph [14], a conventional cloud computing system, which only configures and launches computing instances on demand, may take minutes to provide the requested computing resources. The invention avoids this undesirable delay.

2.4 In order to predict the customer demand, the invention analyses a history of instance requests, for example the number and types of instances that are requested; see figure 5 and [15,56-57]. Based on this prediction, instances are pre-configured which can be quickly activated or allocated to the customer.

2.5 As shown in figure 7 (see 618) and explained in [81-82] the pre-warming of computing instances includes accessing, caching, decrypting and decompressing the machine images and modifying their configuration files.

3. The board's understanding of the invention

3.1 The board leaves open the question of whether claims 1 and 3 of both requests are clear within the meaning of Article 84 EPC, and instead bases its analysis on its understanding of the invention as set out in the summons.

3.2 The invention relates to providing "compute instances", realised as "virtual machines" (VMs) in response to customer requests. Generally speaking, virtual machines are programs which are stored as "machine images" and which, when run, provide the functionality of a physical computer. A machine image is "configured" before it is launched. A VM requires both memory resources and, once launched, CPU (central processing unit) time. The board regards it as implicit in such a cloud computing environment comprising "computing devices" that means are present for determining which of the computing devices are to host the compute instances.

3.3 In claim 1 of both requests the expression "activate the compute instance from one of the plurality of machine images that are idle on the one or more computing devices" is understood to mean "activate the compute instance from one of the plurality of **compute instances** that are idle on the one or more computing devices", since the claim has previously set out idling compute instances.

3.4 Claim 1 sets out the expression, taken from original claim 4, "an expected availability of the plurality of computing devices". Original claim 1 sets out one or more datacentres comprising a plurality of "computing devices". Hence the board understands the "computing devices" to be the computing hardware on which the "compute instances" run. As to the "availability" of the computing devices, the board notes that paragraph [72] of the description gives an example of 20% of the computing resources, understood to be the computing devices, being available for pre-warmed compute instances. According to paragraph [73], the resources

reserved for pre-warmed compute devices may be reduced to 10% due to planned outages or maintenance.

3.5 The board understands balancing "the expected demand with the expected availability to meet a constraint associated with the launching and idling the plurality of compute devices" to mean that the number of pre-warmed compute instances depends on the expected demand for compute instances and the available hardware resources, the number of compute devices in any cloud computing environment being a "constraint".

3.6 In claims 1 and 3 of the auxiliary request the expression "modifying a configuration file of the machine image based at least in part on a configuration of the compute instance" is understood to mean "modifying the configuration of the **machine image** based at least in part on the **configuration file** of the compute instance" (emphasis by the board).

4. Document D1 (Shohei Yamasaki et al., "Model-based resource selection for efficient virtual cluster deployment")

4.1 The decision started from D1 in its assessment of inventive step. D1 relates to a form of distributed computing, termed grid computing, in which clusters of virtual machines (VMs)(see figure 1) are made available on shared hardware in response to user install requests submitted to the "head node"; see abstract. This is achieved by a "Virtual Cluster Installer" (VPC; see page 2, section 2) which responds to said requests by creating VM images. The VPC reduces VM installation time by selecting nodes in increasing order of predicted installation time.

4.2 VM images are frequently requested software that is not yet fully configured. The VPC automatically determines the packages to install in VM images by finding frequently-appearing packages in a user-request history; see page 2, right-hand column, first paragraph. The install requests specify hardware and software requirements, including the CPU type and speed, RAM amount, disk space and a number of nodes; see page 2, section 2.1. The master node contacts the head node for each site (see figure 1; site A and site B) for VM hosting nodes satisfying the requested hardware specification; see page 2, section 2.2.

4.3 According to section 2.3, "on each VM hosting node, a fresh VM instance is running and waiting for the head node's installation request". The VM image and any remaining packages are made available from the head node and then installed. The cluster installer then configures the VM according to the user request. The VM mounts the VM image as its base file system, and the cluster installer performs the configuration specified in the install request.

4.4 According to section 3 on page 3, to model virtual cluster installation a node-level model is created which predicts the installation time at each node. The model divides the installation process into five steps (see page 3, section 3.1): step 1 "package download"; step 2 "package transfer"; step 3 "package installation"; step 4 "configuration" and step 5 "reboot". Step 3 (package installation) of the model assumes that the VM images are already "staged" to each node, which the board understands in the context of cacheing (see page 3, right column, lines 1 to 7) to mean that VM images are cached at each node, thus avoiding transfer delays. Each node caches frequently requested VMs, and the

first use of a different image is said to be a rare "cache miss". In step 4 configuration scripts are run to set up standard system properties such as the host name and IP address. In step 5, after the node has completed installing the user-requested VM image, the VM is rebooted with the customised image, which the board understands to be "launching and idling" the VM as presently claimed.

4.5 According to section 3.2, the virtual cluster installation model calculates the execution time of each of the five steps by taking into account *inter alia* disk I/O performance; see page 4, table 1. Section 4.1 on page 4 contrasts the use of a VM selection policy based on the model with *inter alia* a "DISK" policy based on disk I/O performance in which VM hosting nodes are selected in decreasing order of I/O performance; see figure 4. According to section 4.3 ("Experimental results"), in particular section 4.3.1 and figures 5 and 6, the model VM node selection policy achieved an 11% reduction in installation time compared to the "DISK" policy. The same trend is shown in figures 7 and 8; see section 4.3.2.

4.6 A key disclosure in D1 (page 2, right-hand column, lines 7 to 9) appears to be that the VPC finds frequently-appearing packages in the user-request history and uses this information to determine the packages to install in VM images. The virtual cluster installation model then assumes that each install request is for a VM image with the most frequently requested packages; see page 3, right column, lines 1 to 7. An actual install request specifying other packages, termed a "cache miss" does not substantially increase the installation times achieved by the model-

based installation approach; see page 5, right column, lines 11 to 21.

4.7 Hence, in the terms of claim 1 of the main request, D1 discloses the following features:

A system (see figure 1) for provisioning and launching compute instances, the system comprising: one or more datacenters (site A, site B) comprising a plurality of computing devices configured to communicate with each other over network connections; and one or more memories having stored therein computer-readable instructions (model policy) that, upon execution on the system, cause the system at least to: determine an expected demand (page 2, section 2) for a plurality of compute instances, the plurality of compute instances being associated with a plurality of machine images and a plurality of compute instance types, determine one or more computing devices from the plurality of computing devices configured to host the plurality of compute instances based at least in part on the expected demand, wherein the instructions to determine an expected demand for a plurality of compute instances comprise instructions that, upon execution on the system, cause the system at least to: analyze a history of requests (see page 2, section 2, right column, lines 4 to 10) for compute instances to determine a target for the plurality of compute instances and the one or more computing devices.

4.8 Given the board's understanding regarding the expressions "balancing" and "constraint", the board regards it as implicit in D1 that the cloud computing environment determines an expected availability of the plurality of computing devices and "balances" the expected demand with the expected availability to meet

a constraint associated with launching and idling the plurality of compute instances, the one or more computing devices being determined based at least in part on the balance.

5. Inventive step, Article 56 EPC

5.1 The appealed decision

5.1.1 According to the appealed decision (reasons 4.2),

"The subject-matter of the claims differs in the newly added features at the end of the claim. The expected availability of the computing devices is determined and taken into account when preconfiguring ("pre-warming") the compute instances."

5.1.2 The board finds it regrettable that the decision does not state the difference features more explicitly. Based on a comparison of claim 1 as originally filed and that of the main request, the board understands the decision to find that the subject-matter of claim 1 of the main request differs from the disclosure of D1 in the following features:

"wherein the instructions to determine one or more computing devices from the plurality of computing devices comprise instructions that, upon execution on the system, cause the system at least to: determine an expected availability of the plurality of computing devices; balance the expected demand with the expected availability to meet a constraint associated with launching and idling the plurality of compute instances; and determine the one or more computing devices based at least in part on the balance."

5.1.3 The problem to be solved was regarded as avoiding preconfiguring compute instances that later failed to serve the subsequent customer requests. The claimed solution, namely determining the expected availability of the devices involved, did not involve an inventive step. The skilled person would have made use of such information when choosing the devices to preconfigure, should the information have been available. The claims covered the trivial case in which, for example, a broken computing device was not used for preconfiguring any compute instance. Hence claim 1, amongst others, did not involve an inventive step.

5.2 The grounds of appeal

5.2.1 The appellant has argued that the subject-matter of claim 1 of the main request differs from the disclosure of D1 in more features than stated in the decision, namely:

"determine an expected demand for a plurality of compute instances, the plurality of compute instances being associated with a plurality of machine images and a plurality of compute instance types, wherein the expected demand comprises the number and types of compute instances that are expected to be requested; determine one or more computing devices from the plurality of computing devices configured to host the plurality of compute instances based at least in part on the expected demand, wherein the instructions to determine an expected demand for a plurality of compute instances comprise instructions that, upon execution on the system, cause the system at least to: analyze a history of requests for compute instances to determine a target for the plurality of compute instances and the one or more computing devices; and revise the target

based at least in part on statistical factors associated with the requests; prior to receiving a request from a customer node to launch a compute instance, launch and idle the plurality of compute instances on the one or more computing devices based at least in part on the plurality of machine images and the plurality of compute instance types; and wherein the instructions to determine one or more computing devices from the plurality of computing devices comprise instructions that, upon execution on the system, cause the system at least to: determine an expected availability of the plurality of computing devices; balance the expected demand with the expected availability to meet a constraint associated with launching and idling the plurality of compute instances; and determine the one or more computing devices based at least in part on the balance".

5.2.2 A problem solved by the invention was how to dynamically scale the computing resources available to meet the needs and requirements of the various entities using the services. D1 did not mention predicting a demand, i.e. the numbers and types, of instances that a service provider expected its customers to request. Nor did D1 disclose launching and idling instances prior to receiving a launch request from a customer. Thus there was no hint in D1 that would have led the skilled person to the claimed solution.

5.3 The board's view

5.3.1 As set out in the annex to the summons to oral proceedings, the subject-matter of claim 1 differs from the disclosure of D1 in that the instructions to determine one or more computing devices from the plurality of

computing devices comprise instructions that, upon execution on the system, cause the system at least to:

a. determine an expected demand comprising the number and types of compute instances that are expected to be requested;

b. revise the target based at least in part on statistical factors associated with the requests;

c. prior to receiving a request from a customer node to launch a compute instance, to launch and idle the plurality of compute instances on the one or more computing devices based at least in part on the plurality of machine images and the plurality of compute instance types and

d. in response to receiving the request from the customer node to launch the compute instance, to activate the compute instance from one of the plurality of compute instances that are idle on the one or more computing devices.

5.3.2 Differences "a" and "b" relate to usual matters of data analysis and optimisation to predict a demand, given the disclosure in D1 (page 2, right column, lines 7 to 10) of finding frequently appearing packages in the user-request history and the mention of "statistical linear regression" in the sentence bridging the first two pages.

5.3.3 An inventive step, if any, could lie in differences "c" and "d" which relate to launching and idling compute instances in anticipation of a user request. D1 discloses on page 2, right column, lines 4 to 10, storing pre-configured VM images with the most commonly-reques-

ted packages, as established by analysing the user-request history (see page 3, right column, lines 10 to 11), and satisfying a user request by selecting the most similar stored VM image and fetching any missing packages required by the request from the nearest package repository; see page 2, right column, section 2.3, lines 6 to 7, and page 3, section 3.1, step 1. The user-requested VM image is subsequently "rebooted", which the board understands as launching and idling a VM instance. The question arises what the skilled person starting from D1 would have done in the cases where requests for VM instances could be immediately satisfied using stored VM images "as is", thus avoiding the need to fetch further packages from a local package repository; see page 3, section 3.1, step 1. Given that D1 aims to satisfy requests for VM instances more quickly, the skilled person would have recognised the potential for further saving time by launching and idling the most commonly requested VM instances in advance, thus adding features "c" and "d" as a matter of usual design.

5.3.4 Hence the subject-matter of claim 1 of the main request does not involve an inventive step in view of D1.

6. The auxiliary request

6.1 In the board's understanding (see above), claim 1 sets out the following additional features:

"wherein the machine image is pre-configured for launch on the one or more computing devices based at least in part on: caching the machine image from a root storage device that stores the machine image; modifying a machine image based at least in part on a configuration file of the compute instance; and allocating a storage

volume to the compute instance based at least in part on the configuration of the compute instance."

6.2 According to the decision, these additional features were known from D1; see page 3, top of right-hand column (caching images); page 2, right-hand column, lines 5 to 9, (implicitly disclosing the step of modifying the configuration) and page 4, left-hand column, last paragraph (allocating storage). Hence the additional features were unable to lend inventive step to the independent claims.

6.3 The appellant has argued that, although the top of the right-hand column of page 3 referred to caching a VM image, this part of D1 also stated that "In Step 3, we do not consider the time for VM image transfer", assuming that every VM image was already staged to each node. Therefore, D1 did not disclose a VM image being cached as part of a launch and idle procedure, as in claim 1, since every VM image was already "cached". Furthermore, according to Step 3 of D1 (page 3, left-hand column), each VM hosting node mounted the VM image, and installed transferred packages into the image. Accordingly, the machine images were not modified prior to being launched, since the VM images already staged at each node were mounted without modification. As to the final feature of claim 1 (see above), section 4.1 on page 4, left-hand column of D1 (which includes the final paragraph mentioned in the decision) referred to the size of additional packages that must be dynamically downloaded and disk policies. However section 4.1 of D1 was silent on allocating a storage volume to the compute instance, or any allocation of storage volume. Thus claim 1 of the auxiliary request was further distinguished from D1. D1 taught away from caching the machine image and modifying a

configuration file of the machine image by teaching that every VM image was already staged to each node. This measure was seemingly taken to reduce the time taken to launch a VM after receiving a client instruction. Thus claim 1 of the auxiliary request was inventive with respect to D1.

6.4 The board takes the view that there is explicit disclosure of the "caching" and "modifying" parts of the additional features, but not the final aspect of allocating a storage volume, in D1. The board notes however that in D1 (see page 3, right column, first paragraph) VM images from the "head node" (see figure 3), a "root host" in the sense of the claims, are transferred to each "hosting node" (see section 3.1, step 2) where they are cached. This is another way of saying that the VM images are "staged" to the hosting nodes, so that no transfer time is incurred when installing a new VM instance; see page 3, right column, first sentence.

6.5 D1 also discloses modifying VM images by fetching additional packages from a package repository; see figure 3 and section 3.1, step 1. It is implicit in D1 that the additional packages are specified in the user request, it being a usual measure for the skilled person to structure such data as a "configuration file". Although D1 does not explicitly mention the configuration file also specifying a storage volume for a new VM instance, the skilled person would have understood the need for a VM instance to have a storage volume, specifying the storage volume in the configuration file being an obvious option for the skilled person.

6.6 Hence the board finds that the additional features added to claim 1 are unable to lend inventive step to the claim in view of D1.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

K. Götz-Wein M. Müller

Decision electronically authenticated