(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 4 047 478 A1**

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
24.08.2022 Bulletin 2022/34

(21) Application number: **22305139.2**

(22) Date of filing: **09.02.2022**

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

(52) Cooperative Patent Classification (CPC):
G06F 9/505

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
KH MA MD TN

(30) Priority: **18.02.2021 US 202117178943**

(71) Applicant: **Amadeus S.A.S.**
06410 Biot (FR)

(72) Inventors:
- **GRABARSKY, Philippe**
  34470 Pérols (FR)
- **NSIRI, Mohamed Wadie**
  06110 Le Cannet (FR)

(74) Representative: **Samson & Partner Patentanwälte mbB**
Widenmayerstraße 6
80538 München (DE)

(54) DEVICE, SYSTEM AND METHOD FOR ASSIGNING PORTIONS OF A GLOBAL RESOURCE LIMIT TO APPLICATION ENGINES BASED ON RELATIVE LOAD

(57) A device, system and method for assigning portions of a global resource limit to application engines based on relative load is provided. A system comprises a plurality of application engines that share a global resource limit; and a plurality of operator engines. The plurality of operator engines are each configured to: monitor a respective metric representative of respective load at a respective application engine; share the respective metric with others of the plurality of operator engines; determine a relative load at the respective application engine based on the respective metric and respective metrics received from the others of the plurality of operator engines; and assign a portion of the global resource limit to the respective application engine based on the relative load.

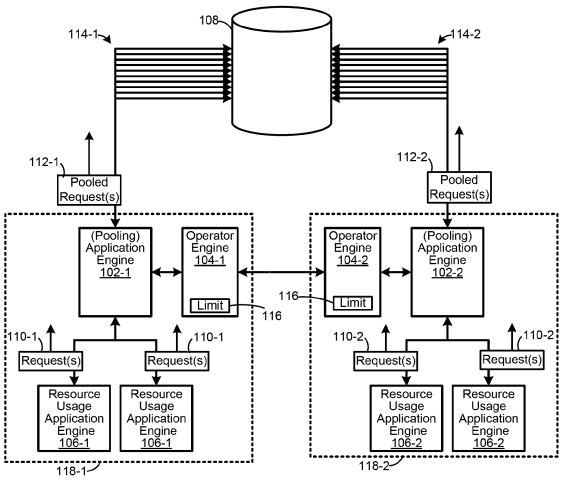

FIG. 1

EP 4 047 478 A1

Description

BACKGROUND

[0001] In container-orchestrated environments (COEs), and the like, associated applications may all be attempting to use a same hardware resource, such as the associated applications requesting database connections with a database. As such, a resource pooling application may be used with associated applications to manage requests to the database. However, as the associated applications, along with an associated pooling application are scaled, for example by adding instances thereof in a container-orchestrated environment, the plurality of instances of the associated applications, and/or the associated pooling application, may result in many requests, and the like, to use the same hardware resource, which may cause operation of the hardware resource to decline and/or slowdown, and the like, which may result in an overall drop in efficiency in processing the attempts to use the hardware resource.

SUMMARY

[0002] An aspect of the present specification provides a system comprising: a plurality of application engines that share a global resource limit; a plurality of operator engines each configured to: monitor a respective metric representative of respective load at a respective application engine; share the respective metric with others of the plurality of operator engines; determine a relative load at the respective application engine based on the respective metric and respective metrics received from the others of the plurality of operator engines; and assign a portion of the global resource limit to the respective application engine based on the relative load.

[0003] At the system of the first aspect, the global resource limit may comprise one or more of a given number of database connections and a hardware-based limit.

[0004] At the system of the first aspect, the plurality of operator engines may each be configured to assign the portion of the global resource limit to the respective application engine based on the relative load by providing a value representative of the portion of the global resource limit to the respective application engine.

[0005] At the system of the first aspect, at least the plurality of application engines may be provided in a container orchestrated environment, such that a number of the plurality of application engines increase and decrease according to demand.

[0006] At the system of the first aspect, the plurality of operator engines may share the respective metric with the others of the plurality of operator engines as a number of the plurality of application engines or the plurality of operator engines increase and decrease.

[0007] At the system of the first aspect, the plurality of operator engines may share the respective metric with the others of the plurality of operator engines as the respective load at the respective application engine changes.

[0008] At the system of the first aspect, the plurality of operator engines may periodically share the respective metric with the others of the plurality of operator engines.

[0009] At the system of the first aspect, the respective metric may comprise transactions per second at the respective application engine.

[0010] At the system of the first aspect, the relative load may be determined by dividing the respective metric by a total of respective metrics of the plurality of operator engines.

[0011] At the system of the first aspect, the global resource limit may comprise a maximum number of database connections to a database, and each of the plurality of application engines may comprise a respective remote database connection pooling engine configured to consolidate database connection requests for respective associated database lookup engines.

[0012] At the system of the first aspect, the system comprises at least two instances of associated application engines and operator engines, wherein one operator engine and at least one application engine with its respective remote database connection pooling engine are associated in one instance, and wherein the instances are implemented at different cloud computing devices.

[0013] Accordingly, the operator can share respective metric across different cloud computing devices. For example, it is thus possible to benefit from the advantage of having a plurality of different cloud computing devices while keeping a balanced metric, e.g. number of connections, across the different cloud computing devices.

[0014] A second aspect of the specification provides a method comprising: monitoring, at an operator engine, a respective metric representative of respective load at a respective application engine, the respective application engine being one of a plurality of application engines that share a global resource limit, the operator engine being one of a plurality of operator engines; sharing, at the operator engine, the respective metric with others of the plurality of operator engines; determining, at the operator engine, a relative load at the respective application engine based on the respective metric and respective metrics received from the others of the plurality of operator engines; and assigning, at the operator engine, a portion of the global resource limit to the respective application engine based on the relative load.

[0015] In some embodiments, the method of the second aspect comprises method steps corresponding to any one of the system of the first aspect functions set out above.

[0016] Notably, at the method of the second aspect, the global resource limit may comprise one or more of a given number of database connections and a hardware-based limit.

[0017] For example, at the method of the second aspect, assigning, at the operator engine, the portion of the global resource limit to the respective application engine

based on the relative load may occur by providing a value representative of the portion of the global resource limit to the respective application engine.

[0018] At the method of the second aspect, at least the plurality of application engines may be provided in a container orchestrated environment, such that a number of the plurality of application engines increase and decrease according to demand.

[0019] At the method of the second aspect, sharing the respective metric with others of the plurality of operator engines may occur as a number of the plurality of application engines or the plurality of operator engines increase and decrease.

[0020] At the method of the second aspect, sharing the respective metric with others of the plurality of operator engines may occur as the respective load at the respective application engine changes.

[0021] At the method of the second aspect, sharing the respective metric with others of the plurality of operator engines may occur periodically.

[0022] At the method of the second aspect, the respective metric may comprise transactions per second at the respective application engine.

[0023] At the method of the second aspect, the relative load may be determined by dividing the respective metric by a total of respective metrics of the plurality of operator engines.

[0024] At the method of the second aspect, the global resource limit may comprise a maximum number of database connections to a database, and each of the plurality of application engines may comprise a respective remote database connection pooling engine configured to consolidate requested database connection requests for respective associated database lookup engines.

[0025] Notably, at the method of the second aspect, one operator engine and at least one application engine with its respective remote database connection pooling engine are associated in one instance and wherein the instances are implemented at different cloud computing devices.

BRIEF DESCRIPTIONS OF THE DRAWINGS

[0026] For a better understanding of the various examples described herein and to show more clearly how they may be carried into effect, reference will now be made, by way of example only, to the accompanying drawings in which:

FIG. 1 depicts a system for assigning portions of a global resource limit to application engines based on relative load, according to non-limiting examples.

FIG. 2 depicts a device for assigning portions of a global resource limit to application engines based on relative load, according to non-limiting examples.

FIG. 3 depicts a method for assigning portions of a global resource limit to application engines based on relative load, according to non-limiting examples.

FIG. 4 depicts operator engines of the system of FIG. 1 exchanging respective metrics representative of respective load at respective application engines, according to non-limiting examples.

FIG. 5 depicts application engines of the system of FIG. 1 throttling usage of a hardware resource according to portions of a global resource limit assigned thereto by the operator engines, the portions based on the relative load, according to non-limiting examples.

DETAILED DESCRIPTION

[0027] Attention is directed to FIG. 1 which depicts a system 100 for assigning portions of a global resource limit to application engines based on relative load.

[0028] The components of the system 100 are generally in communication via communication links which are depicted in FIG. 1, and throughout the present specification, as double-ended arrows between respective components. The communication links includes any suitable combination of wireless and/or wired communication networks and, similarly, the communication links may include any suitable combination of wireless and/or wired links.

[0029] The system 100 will furthermore be described with respect to engines. As used herein, the term "engine" refers to hardware (e.g., a processor, such as a central processing unit (CPU) an integrated circuit or other circuitry) or a combination of hardware and software (e.g., programming such as machine- or processor-executable instructions, commands, or code such as firmware, a device driver, programming, object code, etc. as stored on hardware). Hardware includes a hardware element with no software elements such as an application specific integrated circuit (ASIC), a Field Programmable Gate Array (FPGA), etc. A combination of hardware and software includes software hosted at hardware (e.g., a software module that is stored at a processor-readable memory such as random access memory (RAM), a hard-disk or solid-state drive, resistive memory, or optical media such as a digital versatile disc (DVD), and/or implemented or interpreted by a processor), or hardware and software hosted at hardware.

[0030] A system 100 comprises a plurality of application engines 102-1, 102-2 that share a global resource limit, described in more detail below. The plurality of application engines 102-1, 102-2 are interchangeably referred to herein, collectively, as the application engines 102 and, generically, as an application engine 102. This convention will be used elsewhere in the specification. Furthermore, while the system 100 is described with respect to two application engines 102, the system 100 may comprise any suitable number of application engines 102 that may be larger than "2".

[0031] The system 100 further comprises a plurality of operator engines 104-1, 104-2 (e.g. operator engines 104 and/or an operator engine 104) which, as depicted,

may be in a one-to-one relationship with the plurality of application engines 102; alternatively, one operator engine 104 may be associated with more than one application engine 102. Hence, the system 100 may include a same number of operator engines 104 as application engines 102 (e.g. which may be greater than "2" operator engines 104 and/or application engines 1002), or the system 100 may include a smaller number of operator engines 104 as application engines 102. In general, however, the system 100 includes any suitable number of operator engines 104 and/or application engines 102. Operation of the operator engines 104 are described below.

[0032] As depicted, the system 100 further comprises, for each application engine 102, a plurality of resource usage application engines 106-1 (e.g. associated with the application engine 102-1) or resource usage application engines 106-2 (e.g. associated with the application engine 102-2). The resource usage application engines 106-1, 106-2 are interchangeably referred to herein, collectively, as the resource usage application engines 106 and, generically, as a resource usage application engines 106.

[0033] In particular, the resource usage applications engines 106 may be used to access a hardware resource 108, for example, as depicted, a database.

[0034] As depicted, the application engines 102 may comprise resource pooling engines (e.g. "Pooling" applications in FIG. 1) which, for example, receive respective requests 110 from respective resource usage applications engines 106, and pool the requests 110, to submit to the hardware resource 108 as respective pooled requests 112-1, 112-2 (e.g. pooled requests 112 and/or a pooled request 112) using, for example, connections 114-1, 114-2 (e.g. connections 114 and/or a connection 114) to the hardware resource 108. It is understood that a pooled request 112 generally comprises one or more a request 110 such that a pooled request 112 may comprise combined and/or bundled requests 110 with an application engine 102 configured accordingly. A pooled request 112 may hence be of a format compatible with combining and/or bundling requests 110 (though the requests 110, 112 may be of a same format). Furthermore, a pooled request 112 may comprise combined and/or bundled requests from different resource usage application engines 106.

[0035] In a particular example, components of the system 100 may generally be associated with, and/or operated by an entity, such as a company, and the like, that may provide computer-based services, and in particular computer-based services for the travel industry, via the system 100. For example, terminals (not depicted) at travel agencies, airports, airline offices, and the like, and/or computing devices of consumers, may access the resource usage applications engines 106 via a network, and the like, such as the Internet, for example via a browser, and the like, and/or a special purpose application, and the resource usage applications engines 106 may comprise instances of websites, and the like, for providing search fields, and the like, for performing searches for travel information (e.g. plane schedules, availability on particular routes, and the like, though such travel information may include any suitable travel information associated with hotels, trains, buses, car rentals, and the like). A request 110 may hence comprise a request for travel information, and the like, which may be used to search the database of the hardware resource 108, which may comprise a database of travel information and the like. As many instances of the resource usage applications engines 106 may be active at any given time, the requests 110 may be pooled by the application engines 102 as the pooled requests 112, and a number of the requests 110 may be in the millions or higher, though the number of the requests 110 may be any suitable number.

[0036] While not depicted, the database of the hardware resource 108 is understood to comprise one or more servers and/or computing devices, and the like, which store information of the database, as well as process and return, to the application engines 102 responses to the pooled requests 112; the application engines 102 may then return a response from the hardware resource to a respective resource usage application engines 106 which originated a request 110 that resulted in the response. For example, such response may include travel information requested via a request 110.

[0037] However, the hardware resource 108 may have an associated global resource limit 116 which may include, but is not limited to, a maximum number of the connections 114.

[0038] Hence, in examples where the hardware resource 108 comprises a database, and the global resource limit 116 comprises a maximum number of database connections 114 to the database, requests 110 may comprise database lookup requests, and the resource usage applications engines 106 may comprise database lookup engines, each of the plurality of application engines 102 may comprises a respective remote database connection pooling engine which pools database lookup requests from the database lookup engines.

[0039] For example, a database (e.g. hardware of a computing device implementing the database) may be able to process (and/or handle, and the like), only a given number of connections 114 at any given time. As such, when more connections 114 are needed to process the pooled requests 112 than are available, and/or the application engines 102 may attempt to open more connections 114 than are available, the database of the hardware resource 108 may operate inefficiently which may slow processing of received pooled requests 112, which may lead to an overall slowdown of the system 100.

[0040] However, while present examples are described with respect to a database and/or connections 114 thereto, in other examples, the global resource limit 116 may comprise any suitable hardware-based limit. For example, the application engines 102 may attempt to access hardware based ports at a computing device

(e.g. of the hardware resource 108) and the global resource limit 116 may comprise a number of such ports. Indeed, a maximum number of connections 114 may also be understood to be hardware based, as many database systems are limited to a given number of connections per processor core being used to implement a database. As such, the global resource limit 116 may comprises one or more of a given number of database connections and/or any other suitable hardware-based limit.

[0041] As such, the operator engines 104 are preconfigured with the global resource limit 116 (e.g. stored at respective memories thereof) and the operator engines 104 may be configured to: monitor a respective metric representative of respective load at a respective application engine 102; share the respective metric with others of the plurality of operator engines 104; determine a relative load at the respective application engine 102 based on the respective metric and respective metrics received from the others of the plurality of operator engines 104; and assign a portion of the global resource limit 116 to the respective application engine 102 based on the relative load. Such assignment will be described in further detail below, however, in general, an operator engine may provide an indication of a determined portion of the global resource limit 116 to a respective application engine 102 (e.g. via a communication link therebetween, and the like) such that the respective application engine 102 limits itself to using the determined portion of the global resource limit 116.

[0042] Hence, as depicted, the operator engines 104 are in communication with a respective application engine 102, and also in communication with other operator engines.

[0043] Prior to further discussing operation of the operator engines 104, further aspects of the system 100 are next described.

[0044] In particular, the application engines 102, the resource usage engines 106 (and optionally the operator engines 104) may be implemented in a container-orchestrated environment (COE) in which different instances 118-1, 118-2 (e.g. instances 118 and/or an instance 118, and which may be larger than two instances 118) of associated application engines 106, operator engines 104, and resource usage engines 106 may increase or decrease based on demand. Put another way, the plurality of application engines 102 (and optionally the plurality of operator engines 104) (e.g., and similarly the associated application engines 106) may be provided in a container-orchestrated environment, such that a number of the plurality of application engines 102 (and optionally the plurality of operator engines 104) (e.g., and similarly the associated application engines 106) may increase and decrease according to demand. However, while the operator engines 104 are depicted as being a component of a respective instance 118, and hence may scale up or down with the instances, in other examples the operator engines 104 may not scale up or down and/or may scale up or down independent of the instances 118 such that one operator engine 104 may: monitor a respective metric representative of respective load at a plurality of respective application engines 102; share the respective metric of the respective load of the plurality of respective application engines 102 with others of the plurality of operator engines 104; determine respective relative loads of the plurality of respective application engines 102 based on the respective metrics, and other respective metrics received from the others of the plurality of operator engines 104; and assign respective portions of the global resource limit 116 to the plurality of respective application engines 102 based on the respective relative loads.

[0045] Hence, while not depicted, the system 100 may include a container-orchestrated environment (COE) engine, and the like, to manage a number of the instances 118 as demand for the application engines 102 and/or the resource usage engines 106 changes. Such a COE engine may generally execute a plurality of instances of an application, which may change (e.g. increase or decrease) as demand changes, which may be referred to as auto-scaling of an application.

[0046] Hence, while in the depicted example there are at least two instances 118 of associated application engines 102, operator engines 104, and resource usage engines 106 are depicted, a number of the instances 118 may increase (e.g. to greater than two) or decrease (e.g. to at least two).

[0047] Alternatively, and/or in addition, the application engines 102 and the resource usage engines 106 may be implemented in a Platform-as-a-service (PaaS) environment, which may be implemented via a COE environment. The operator engines 104 may be also be implemented in a PaaS environment which may be a same, or different, PaaS environment as the application engines 102 and the resource usage engines 106.

[0048] In some examples, while the two resource usage engines 106 are depicted for each instance 118, in a COE, number of the resource usage engines 106 for a particular instance may also increase or decrease as demand changes.

[0049] Hence, as depicted, while two resource usage application engines 106 are depicted as being in communication with each application engine 102, as few as one resource usage application engine 106 may be in communication with a respective application engine 102, or more than two resource usage application engines 106 may be in communication with a respective application engine 102.

[0050] In yet further examples, functionality of the application engine 102 and respective resource usage application engines 106 may be combined.

[0051] Regardless, of the number of instances 118, etc., and the number of operator engines 104, it is understood that the operator engines 104 are all in communication with each other such that the operator engines 104 may share, with each other, metrics of respective application engines 102 which represent respective

load on respective application engines 102, such as a number of requests 110 per unit time that is being received at a respective application engine 102, among other possibilities, including, but not limited to, a number of transactions per second at a respective application engine 102. However, it is understood that the operator engines 104 may communicate with each other in any suitable manner which may include transmission of metrics of respective application engines 102 to other operator engines 104 via suitable communication links, storage of such metrics at one or more databases accessible by the operator engines 104 and/or any other suitable process for communicating. ,

[0052] For example, a transaction may comprise a request 110 and/or, more generically, a transaction may comprise any suitable demand (including, but not limited to, a demand and/or request for usage of hardware resources) placed on a respective application engine 102, for example by the resource usage application engines 106.

[0053] However, other metrics of respective application engines 102 which represent respective load on respective application engines 102 are within the scope of the present specification including, but not limited to, a number of pooled requests 112, and the like, in a queue, an average wait time of a request 112, and the like, in a queue (e.g. before being transmitted to the hardware resource 108), and the like. For example, as a number of transactions per second increase, a number of number of pooled requests 112, and the like, increases, and/or as an average wait time of a request 112, and the like, in a queue increases, a load at a respective application engines 102 is understood to increase.

[0054] Hence, in general, a respective metric of a first application engine 102 may be compared to a same type of respective metric of a second application engine 102 to determine relative load therebetween, and/or a respective metric of a first application engine 102 may be compared to a total of the respective metrics of all the application engine 102 to determine relative load therebetween.

[0055] In yet further examples, a combination of respective metrics may be used, and/or such respective metrics may be combined in weighted manner to determine relative load between the application engines 102.

[0056] Regardless of a type of a respective metric representative of respective load at a respective application engine 102, the operator engines 104 may share such respective metrics (i.e. the operator engines 104 are understood to share a same type of respective metric). A given operator engine 104 may determine relative load of a respective application engine 102, relative to other respective application engine 102, and assign a portion of the global resource limit 116 to the respective application engine 102 based on the relative load.

[0057] In a particular example, global resource limit 116 may comprise 1000 connections 114 to the database of the hardware resource 108. As depicted in FIG. 1, such connections 114 are equally distributed between the application engines 102 (e.g. each of the application engines 102 may use 50% of the connections 114). However, the operator engine 104-1 may determine that the load at the application engine 102-1 is 1500 TPS (e.g. transaction per second), and share this value with the operator engine 104-2. Similarly, the operator engine 104-2 may determine that the load at the application engine 102-2 is 500 TPS (e.g. transaction per second), and share this value with the operator engine 104-1. Hence, the load at the application engine 102-1 is understood to be three times higher than the load at the application engine 102-2. As such, there may be a deficit of connections 114-1 at the application engine 102-1 and a surplus of connections 114-2 at the application engine 102-2.

[0058] As such, each of the operator engines 104 may determine that the total load on the application engines 102 is 2000 TPS (e.g. 1500 TPS plus 500 TPS). The operator engine 104-1 may use the total load, and the load of 1500 TPS at the application engine 102-1, to determine that the relative load on the application engine 102-1 is 0.75 (e.g. 1500 TPS divided by 2000 TPS). Similarly, the operator engine 104-2 may use the total load, and the load of 500 TPS at the application engine 102-2, to determine that the relative load on the application engine 102-2 is 0.25 (e.g. 500 TPS divided by 2000 TPS).

[0059] Continuing with this example, as mentioned above, the global resource limit 116 may comprise 1000 database connections 114. Hence the operator engine 104-1 may assign a portion of 0.75 of the 1000 database connections 114 of the global resource limit 116 to the respective application engine 102-1, or 750 database connections 114; thereafter, the respective application engine 102-1 will attempt to use a maximum of 750 database connections 114 to transmit the pooled requests 112-1 to the database of the hardware resource 108. Similarly, the operator engine 104-2 may assign a portion of 0.25 of the 1000 database connections 114 of the global resource limit 116 to the respective application engine 102-2, or 250 database connections 114; thereafter, the respective application engine 102-2 will attempt to use a maximum of 250 database connections 114 to transmit the pooled requests 112-2 to the database of the hardware resource 108.

[0060] Hence, as illustrated by this example, the relative load of a respective application engine 102 may be determined at an associated operator engine 104, by dividing the respective metric, represented of load of the respective application engine 102 as determined by the associated operator engine 104, by a total of respective metrics of the plurality of operator engines 104.

[0061] In a particular example, to assign a portion of the global resource limit 116 to a respective application engine 102, an operator engine 104 may provide a value representative of the portion of the global resource limit 116 to the respective application engine 102. Continuing with the above example, the operator engine 104-1 may provide a value of "750" database connections 114 to

the application engine 102-1, and the operator engine 104-2 may provide a value of "250" database connections 114 to the application engine 102-2. Thereafter, the application engines 102 may limit their respective number of connections 114 to a respective received value. Put another way, the application engines 102 may throttle usage of hardware at the hardware resource 108, for example by limiting a number of respective connections 114 to the value representative of a portion of the global resource limit 116, and/or limiting usage of hardware at the hardware resource 108 in any other suitable manner as represented by the value representative of a portion of the global resource limit 116.

[0062] Hence, it is further understood that the operator engines 104 generally periodically and/or constantly monitor a respective metric representative of respective load at a respective application engine 102 and, as respective load at a respective application engine 102 changes, the respective metric will change.

[0063] As such, the plurality of operator engines 104 may share the respective metric with the others of the plurality of operator engines 104, as the respective load at the respective application engine 102 changes and/or the respective metric changes. In this manner the operator engines 104 may assign an updated portion of the global resource limit 116 to a respective application engine 102 based on the changed relative load and/or the changed relative metric. In other words, changes to the respective metric at a respective application engine 102 may trigger an operator engine 104 to share the respective metric (e.g. the changed respective metric), which may trigger other operator engines 104, which receive the respective metric, to again share their respective metrics, causing all the operator engines 104 to again assign a portion of the global resource limit 116 to respective application engine 102 based on the relative load.

[0064] Alternatively, the plurality of operator engines 104 (e.g. all of the plurality of operator engines 104) may periodically share the respective metric with the others of the plurality of operator engines 104 which, when a respective metric determined by one or more operator engines 104 has changed, may cause all the operator engines 104 to again assign a portion of the global resource limit 116 to respective application engine 102 based on the relative load (e.g. an updated relative load). However, when no respective metric has changed, then the portion of the global resource limit 116 assigned to a respective application engine 102 is understood not to change.

[0065] In yet further examples, as has already been described, the number of the plurality of application engines 102 and/or the number of operator engines 104 (e.g. and/or the number of instances 188), may increase and decrease. As such, when a new application engine 102 and/or a new operator engine 104 is provided in the system 100, the portions of the global resource limit 116 assigned to the respective application engines 102 is understood to change as the new application engine 102 is understood to require a portion of the global resource limit 116 and/or a new operator engine 104 may be "managing" new application engines 102 and/or management of a portion of existing applications engines 102 may be transferred to the new operator engine 104 (e.g. from another existing operator engine 104). Similarly, when an existing application engine 102 and/or an operator engine 104 is removed from the system 100, the portions of the global resource limit 116 assigned to a respective application engines 102 is understood to change, as the load may be redistributed among the remaining application engines 102, and/or management of a portion of existing applications engines 102 may be transferred from the operator engine 104 being removed to other operator engines 104. As such, in some of these examples, the plurality of operator engines 104 may share the respective metric with others of the plurality of operator engines 104 as a number of the plurality of application engines 102 and/or a number of operator engines 104 increase and decrease, which again causes assigning of a portion of the global resource limit 116 to the respective application engines 102 based on the relative load.

[0066] In particular, as mentioned above, without the operator engines 104 communicating such metrics with each other, the application engines 102 may attempt to use any number of connections 114 which, in total, may exceed the maximum number represented by the global resource limit 116. Alternatively, the application engines 102 may be arbitrarily assigned a relative number of connections 114, such as the maximum number represented by the global resource limit 116 divided by the number of the instances 118 and/or the number of the application engines 102 (e.g. as depicted, 50% of the connections 114 to each of the application engines 102), which may cause a surplus of connections 114 at one application engine 102 with low relative load, and a deficit of connections 114 at another application engine 102 with high relative load. Hence, in general, the operator engines 104 attempt to dynamically balance usage of the connections 114 by the application engine 102 based on relative load.

[0067] Attention is next directed to Fig. 2 which depicts a block diagram of an example device 200 that includes a controller 202 communicatively coupled to a memory 204 and a communication interface 206. The device 200 may be generally configured to implement the engines 102, 104, 106 of the system 100, as well as numbers thereof, and/or numbers of the instances 118. It is furthermore understood that the device 200 may be implemented as one or more servers and/or one or more cloud computing devices, with functionality thereof distributed across one or more servers and/or one or more cloud computing devices. As such, instances 118 may be implemented at different cloud computing devices in communication with each other, for example distributed geographically, and which may coordinate implementation of a containerized-orchestrated environment, and/or coordinate implementation of platform-as-a-service envi-

ronments.

[0068] The controller 202 comprise one or more general-purpose processors and/or one or more special purpose logic devices, such as microprocessors (e.g., a central processing unit, a graphics processing unit, etc.), a digital signal processor, a microcontroller, an ASIC, an FPGA, a PAL (programmable array logic), a PLA (programmable logic array), a PLD (programmable logic device), etc.

[0069] The controller 202 is interconnected with the memory 204 which may comprise any suitable memory that stores instructions, for example, as depicted, in the form of modules, described below, that, when implemented by the controller 202, cause the controller 202 to implement the engines 102, 104, 106 and/or the instances 118. The memory 204 may be implemented as a suitable non-transitory computer-readable medium (e.g. a suitable combination of non-volatile and volatile memory subsystems including any one or more of Random Access Memory (RAM), read only memory (ROM), Electrically Erasable Programmable Read Only Memory (EEPROM), flash memory, magnetic computer storage, and the like). The controller 202 and the memory 204 may be generally comprised of one or more integrated circuits (ICs).

[0070] The controller 202 is also interconnected with a communication interface 206, which generally enables the device 200 to communicate with the other components of the system 100 via one or more communication links. The communication interface 206 therefore includes any necessary components (e.g. network interface controllers (NICs), radio units, and the like) to communicate with the other components of the system 100 via one or more communication links (e.g. via one or more communication networks). The specific components of the communication interface 206 may be selected based on upon types of the communication links. The device 200 may also include input and output devices connected to the controller 202, such as keyboards, pointing devices, display screens, and the like (not shown).

[0071] The memory 204 includes modules. As used herein, a "module" (in some examples referred to as a "software module") is a set of instructions that when implemented or interpreted by a controller and/or a processor, or stored at a processor-readable medium realizes a component or performs a method.

[0072] As depicted, the memory 204 includes various modules 212, 214, 216 which respectively correspond to functionality of the engines 102, 104, 106 of the system 100. For example, the controller 202 may implement the application module 212 to implement one or more application engines 102, the controller 202 may implement the operator module 214 to implement one or more operator engines 104 and/or the controller 202 may implement the resource usage application module 216 to implement one or more resource usage application engines 106. While not depicted, the memory 204 may further store a COE module for implementing a COE engine to increase or decrease the instances 118, and the like, based on demand, as described herein.

[0073] Attention is now directed to FIG. 3 which depicts a flowchart representative of a method 300 for assigning portions of a global resource limit to application engines based on relative load. The operations of the method 300 of FIG. 3 correspond to machine readable instructions that are executed by the device 200 (e.g. and/or by one or more cloud computing devices), and specifically the controller 202 of the device 200 (and/or by controllers of one or more cloud computing devices). In the illustrated example, the instructions represented by the blocks of FIG. 3 may be stored at the memory 204 for example, at least in part as the operator module 214, though other aspects of the method 300 may be implemented via the other modules 212, 216. The method 300 of FIG. 3 is one way in which the device 200, and/or the controller 202 and/or the system 100 may be configured. However, while the method 300 is specifically described with regards to being implemented by the controller 202 and/or the device 200 and/or an operator engine 104 and/or other engines described herein, it is understood that the method 300 may be implemented by one or more cloud computing devices and/or one or more controllers thereof.

[0074] Furthermore, the following discussion of the method 300 of FIG. 3 will lead to a further understanding of the system 100, and its various components.

[0075] The method 300 of FIG. 3 need not be performed in the exact sequence as shown and likewise various blocks may be performed in parallel rather than in sequence. Accordingly, the elements of method 300 are referred to herein as "blocks" rather than "steps." The method 300 of FIG. 3 may be implemented on variations of the system 100 of FIG. 1, as well.

[0076] At a block 302, the controller 202 and/or the device 200 and/or an operator engine 104 monitors a respective metric representative of respective load at a respective application engine 102. For example, an operator engine 104 may periodically query a respective application engine 102 for a respective metric, as described above, and/or a respective application engine 102 may periodically provide the respective metric to a respective operator engine 104, and/or a respective application engine 102 may provide the respective metric to a respective operator engine 104 when the respective metric changes, and the like, among other possibilities.

[0077] As previously mentioned, in a specific example, the respective metric may comprise transactions per second at the respective application engine 102, and/or any other suitable metric of respective load at the respective application engine 102, as described herein.

[0078] At a block 304, the controller 202 and/or the device 200 and/or the operator engine 104 shares the respective metric with others of the plurality of operator engines 104. For example, an operator engine 104 may transmit the respective metric to others of the plurality of operator engines 104 via respective communication links

and/or via any other suitable mechanism, such as mechanisms used within container orchestrated environments.

[0079] At a block 306, the controller 202 and/or the device 200 and/or the operator engine 104 determines a relative load at the respective application engine 102 based on the respective metric and respective metrics received from the others of the plurality of operator engines 104.

[0080] As previously mentioned, in a specific example, the relative load may be determined by dividing the respective metric by a total of respective metrics of the plurality of operator engines 104. However, the relative load may be determined using the respective metric and respective metrics received from the others of the plurality of operator engines 104 in any suitable manner. For example, ratios between the respective metric and the respective metrics received from the others of the plurality of operator engines 104 may used to determine relative load; regardless, however, the relative load at the respective application engine 102, as based on the respective metric and respective metrics received from the others of the plurality of operator engines 104, is understood to be relative to a total load at the application engines 102.

[0081] At a block 308, the controller 202 and/or the device 200 and/or the operator engine 104 assigns a portion of the global resource limit 116 to the respective application engine 102 based on the relative load.

[0082] As has already been described, at the method 300, the operator engine 104 may be configured to assign the portion of the global resource limit 116 to the respective application engine 102 based on the relative load by providing a value representative of the portion of the global resource limit 116 to the respective application engine 102.

[0083] As has already been described, at the method 300, the global resource limit 116 may comprise one or more of a given number of database connections 114 and a hardware-based limit.

[0084] Hence, a respective application engine 102, upon receiving, from an operator engine 104, a value representative of a portion of the global resource limit 116, may responsively throttle usage of hardware at the hardware resource 108, for example by limiting a number of respective connections 114 to the value representative of a portion of the global resource limit 116, and/or limiting usage of hardware at the hardware resource 108 in any other suitable manner as represented by the value representative of a portion of the global resource limit 116.

[0085] Other features described herein may be implemented via the method 300. For example, the plurality of application engines 102 and the plurality of operator engines 104 may be provided in a container orchestrated environment, such that the method 300 may further comprise the controller 202 and/or the device 200 and/or a COE engine increasing and decreasing a number of the plurality of application engines 102 and the plurality of operator engines 104 according to demand.

[0086] In other examples, the method 300 may further comprise the controller 202 and/or the device 200 and/or the operator engine 104 sharing the respective metric (e.g. at the block 304) with the others of the plurality of operator engines 104 as a number of the plurality of operator engines 104 increase and decrease.

[0087] In other examples, the method 300 may further comprise the controller 202 and/or the device 200 and/or the operator engine 104 sharing the respective metric e.g. at the block 304) with the others of the plurality of operator engines 104 as the respective load at the respective application engine 102 changes.

[0088] In yet further examples, the method 300 may further comprise the controller 202 and/or the device 200 and/or the operator engine 104 periodically sharing the respective metric e.g. at the block 304) with the others of the plurality of operator engines 104.

[0089] A specific example of the method 300 will next be described with respect to FIG. 4 and FIG. 5, which are substantially similar to FIG. 1, with like components having like numbers.

[0090] With attention first directed to FIG. 4, the operator engines 104 are understood to be monitoring (e.g. at the block 302 of the method 300) a respective metric 402-1, 402-2 (e.g. a metric 402) representative of respective load at a respective application engine 102, for example by receiving the metric 402 from a respective application engine 102 on request and/or as initiated by a respective application engine 102. For example, the metric 402-1 received at the operator engine 104-1 may comprise 1500 transactions per second at the application engine 102-1, and the metric 402-2 received at the operator engine 104-2 may comprise 500 transactions per second at the application engine 102-2.

[0091] As also depicted in FIG. 4, the operator engines 104 share (e.g. at the block 304 of the method 300) a respective metric 402 with other operator engines 104. For example, as depicted, the operator engine 104-1 provides the metric 402-1 of "1500" transactions per second with the operator engine 104-2, and the operator engine 104-2 provides the metric 402-2 of "500" transactions per second with the operator engine 104-1.

[0092] As depicted in FIG. 5, the operator engines 104 determine (e.g. at the block 306 of the method 300) a relative load 502-1, 502-2 (e.g. relative load 502) at the respective application engines 102 based on the respective metric 402 and respective metrics 402 received from the others of the plurality of operator engines 104.

[0093] For example, as depicted, the operator engine 104-1 divides the metric 402-1 of "1500" with a total of the metric 402-1 of "1500" and the metric 402-2 of "500", for example to determine that the relative load 502-1 at the application engine 102-1 is "0.75" of the total load (e.g. 1500/(1500+500)=1500/2000=0.75).

[0094] Similarly, as depicted, the operator engine 104-2 divides the metric 402-2 of "500" with a total of the metric 402-1 of "1500" and the metric 402-2 of "500", for

example to determine that the relative load 502-2 at the application engine 102-2 is "0.25" of the total load (e.g. 500/(1500+500)=500/2000=0.25).

[0095] The operator engines 104 then assign (e.g. at the block 308 of the method 300) a portion 504-1, 504-2 (e.g. portion 504) of the global resource limit 116 to a respective application engine 102 based on the relative load 502.

[0096] For example, assuming that the global resource limit 116 comprises 1000 connections 114, as depicted, the operator engine 104-1 multiples the relative load 502-1 of "0.75" of the application engine 102-1 by the global resource limit 116 to determine that the portion 504-1 of the connections 114 of the application engine 102-1 is "750" connections.

[0097] Similarly, as depicted, the operator engine 104-2 multiples the relative load 502-2 of "0.25" of the application engine 102-2 by the global resource limit 116 to determine that the portion 504-2 of the connections 114 of the application engine 102-2 is "250" connections.

[0098] The operator engines 104 provide the portions 504 to their respective application engines 102, and the application engines 102 respond by adjusting their connections accordingly. For example, as depicted, the application engine 102-1 increases a number of the connections 114-1 to 750, and the application engine 102-2 increases a number of the connections 114-2 to 250.

[0099] It is understood that the process shown in FIG. 4 and FIG. may be repeated as load at the application engines 104 change such that the number of connections 114 used by a respective application engine 104 may generally correspond to a relative load thereof.

[0100] As should by now be apparent, the operations and functions of the devices described herein are sufficiently complex as to require their implementation on a computer system, and cannot be performed, as a practical matter, in the human mind. In particular, computing devices, and the lie, such as set forth herein are understood as requiring and providing speed and accuracy and complexity management that are not obtainable by human mental steps, in addition to the inherently digital nature of such operations (e.g., a human mind cannot interface directly with digital projectors, digital cameras, RAM or other digital storage, cannot transmit or receive electronic messages, such as a requests and/or the information exchanged between the engines described herein, among other features and functions set forth herein).

[0101] In this specification, elements may be described as "configured to" perform one or more functions or "configured for" such functions. In general, an element that is configured to perform or configured for performing a function is enabled to perform the function, or is suitable for performing the function, or is adapted to perform the function, or is operable to perform the function, or is otherwise capable of performing the function.

[0102] It is understood that for the purpose of this specification, language of "at least one of X, Y, and Z" and "one or more of X, Y and Z" can be construed as X only, Y only, Z only, or any combination of two or more items X, Y, and Z (e.g., XYZ, XY, YZ, XZ, and the like). Similar logic can be applied for two or more items in any occurrence of "at least one..." and "one or more..." language.

[0103] The terms "about", "substantially", "essentially", "approximately", and the like, are defined as being "close to", for example as understood by persons of skill in the art. In some examples, the terms are understood to be "within 10%," in other examples, "within 5%", in yet further examples, "within 1%", and in yet further examples "within 0.5%".

[0104] Persons skilled in the art will appreciate that in some examples, the functionality of devices and/or methods and/or processes described herein can be implemented using pre-programmed hardware or firmware elements (e.g., application specific integrated circuits (ASICs), electrically erasable programmable read-only memories (EEPROMs), etc.), or other related components. In other examples, the functionality of the devices and/or methods and/or processes described herein can be achieved using a computing apparatus that has access to a code memory (not shown) which stores computer-readable program code for operation of the computing apparatus. The computer-readable program code could be stored on a computer readable storage medium which is fixed, tangible and readable directly by these components, (e.g., removable diskette, CD-ROM, ROM, fixed disk, USB drive). Furthermore, it is appreciated that the computer-readable program can be stored as a computer program product comprising a computer usable medium. Further, a persistent storage device can comprise the computer readable program code. It is yet further appreciated that the computer-readable program code and/or computer usable medium can comprise a non-transitory computer-readable program code and/or non-transitory computer usable medium. Alternatively, the computer-readable program code could be stored remotely but transmittable to these components via a modem or other interface device connected to a network (including, without limitation, the Internet) over a transmission medium. The transmission medium can be either a non-mobile medium (e.g., optical and/or digital and/or analog communications lines) or a mobile medium (e.g., microwave, infrared, free-space optical or other transmission schemes) or a combination thereof.

[0105] Persons skilled in the art will appreciate that there are yet more alternative examples and modifications possible, and that the above examples are only illustrations of one or more examples. The scope, therefore, is only to be limited by the claims appended hereto.

Claims

1. A system (100) comprising:

a plurality of application engines (102-1, 106-1,

102-2, 106-2) that share a global resource limit; a plurality of operator engines (104-1, 104-2) each configured to:

monitor a respective metric (402-1, 402-2) representative of respective load at a respective application engine (102-1, 102-2);
share the respective metric with others of the plurality of operator engines (104-2, 104-1);
determine a relative load at the respective application engine (102-1, 102-2) based on the respective metric and respective metrics (402-1, 402-2) received from the others of the plurality of operator engines (104-2, 104-1); and
assign a portion of the global resource limit to the respective application engine based on the relative load.

2. The system of claim 1, wherein the global resource limit comprises one or more of a given number of database connections and a hardware-based limit.

3. The system of claim 1 or 2, wherein the plurality of operator engines (104-1, 104-2) are each configured to assign the portion of the global resource limit to the respective application engine (102-1, 102-2) based on the relative load by providing a value representative of the portion of the global resource limit to the respective application engine.

4. The system of at least one of the claims 1 to 3, wherein at least the plurality of application engines (102-1, 106-1, 102-2, 106-2) are provided in a container orchestrated environment, such that a number of the plurality of application engines increase and decrease according to demand.

5. The system of at least one of the claims 1 to 4, wherein the plurality of operator engines (104-1, 104-2) share the respective metric with the others of the plurality of operator engines (104-2, 104-1) as a number of the plurality of application engines (102-1, 106-1, 102-2, 106-2) or the plurality of operator engines (104-1, 104-2) increase and decrease.

6. The system of at least one of the claims 1 to 5, wherein the plurality of operator engines (104-1, 104-2) share the respective metric with the others of the plurality of operator engines (104-2, 104-1) as the respective load at the respective application engine changes (102-1, 102-2).

7. The system of at least one of the claims 1 to 6, wherein the plurality of operator engines (104-1, 104-2) periodically share the respective metric with the others of the plurality of operator engines (104-2, 104-1).

8. The system of at least one of the claims 1 to7, wherein the respective metric comprises transactions per second at the respective application engine.

9. The system of at least one of the claims 1 to 8, wherein the relative load is determined by dividing the respective metric by a total of respective metrics of the plurality of operator engines (104-1, 104-2).

10. The system of at least one of the claims 1 to 9, wherein the global resource limit comprises a maximum number of database connections to a database (108), and each of the plurality of application engines (102-1, 106-1, 102-2, 106-2) comprises a respective remote database connection pooling engine (102-1, 102-2) configured to consolidate requested database connection requests for respective associated database lookup engines.

11. The system of claim 10, comprising at least two instances (118-1, 118-2) of associated application engines and operator engines, wherein one operator engine (104-1, 104-2) and at least one application engine (106-1, 106-2) with its respective remote database connection pooling engine (102-1, 102-2) are associated in one instance (118-1, 118-2), and wherein the instances are implemented at different cloud computing devices.

12. A method comprising:

monitoring (302), at an operator engine (104-1, 104-2), a respective metric (402-1, 402-2) representative of respective load at a respective application engine, the respective application engine (102-1, 106-1, 102-2, 106-2) being one of a plurality of application engines that share a global resource limit, the operator engine being one of a plurality of operator engines;
sharing (304), at the operator engine (104-1, 104-2), the respective metric with others of the plurality of operator engines;
determining (306), at the operator engine, a relative load at the respective application engine (102-1, 102-2) based on the respective metric and respective metrics received from the others of the plurality of operator engines (104-2, 104-1); and
assigning (308), at the operator engine, a portion of the global resource limit to the respective application engine based on the relative load.

13. The method of claim 12, wherein at least the plurality of application engines (102-1, 106-1, 102-2, 106-2) are provided in a container orchestrated environment, such that a number of the plurality of applica-

tion engines increase and decrease according to demand.

14. The method of claim 12 or 13, wherein the global resource limit comprises a maximum number of database connections to a database (108), and each of the plurality of application engines (102-1, 106-1, 102-2, 106-2) comprises a respective remote database connection pooling engine (102-1, 102-2) configured to consolidate requested database connection requests for respective associated database lookup engines.

15. The method of claim 14, wherein one operator engine (104-1, 104-2) and at least one application engine (106-1, 106-2) with its respective remote database connection pooling engine (102-1, 102-2) are associated in one instance (118-1, 118-2) and wherein the instances (118-1, 118-2) are implemented at different cloud computing devices.

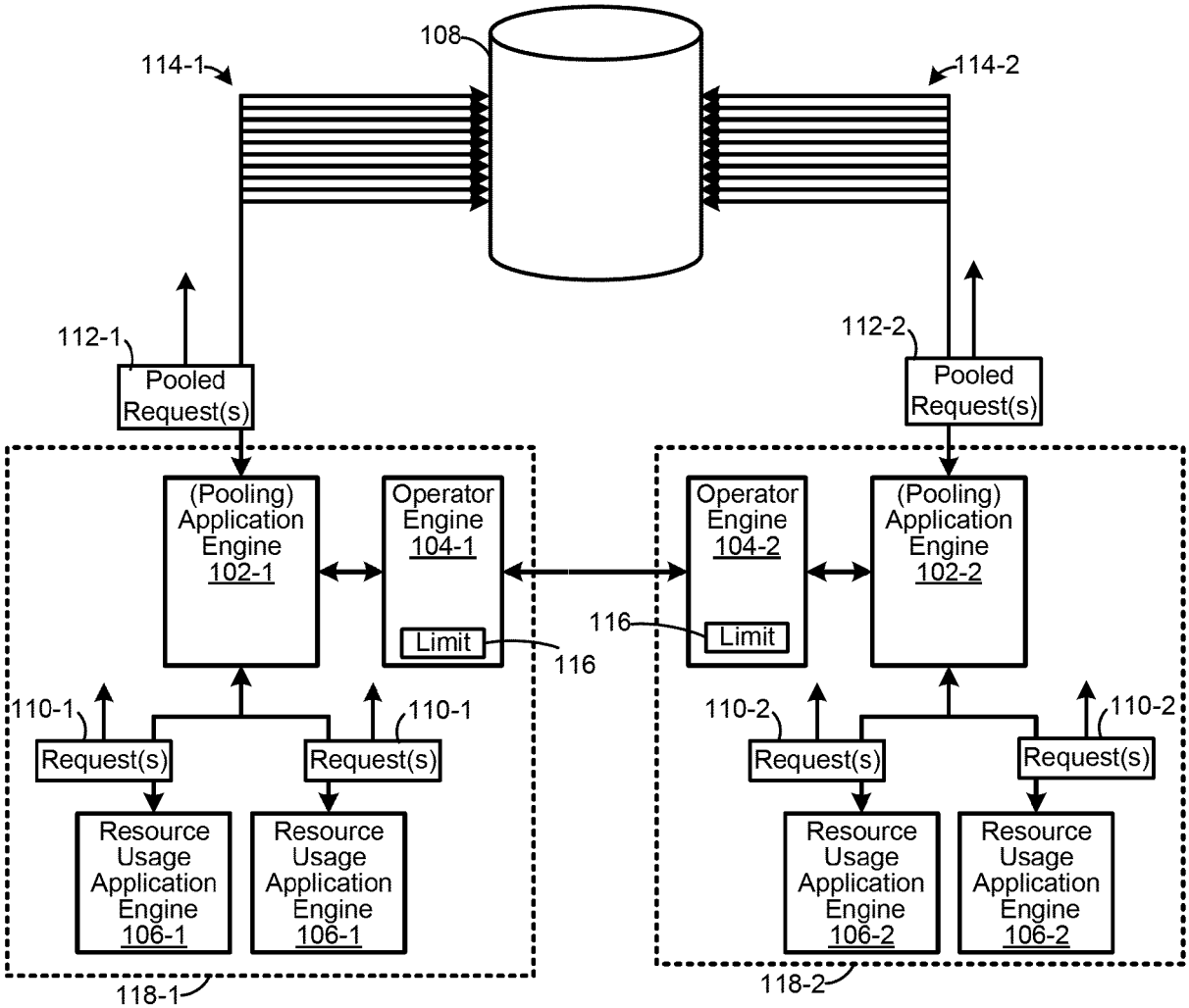
108
114-1
114-2
112-1
Pooled Request(s)
112-2
Pooled Request(s)
(Pooling) Application Engine 102-1
Operator Engine 104-1
Limit
116
Operator Engine 104-2
Limit
116
(Pooling) Application Engine 102-2
110-1
Request(s)
110-1
Request(s)
110-2
Request(s)
110-2
Request(s)
Resource Usage Application Engine 106-1
Resource Usage Application Engine 106-1
Resource Usage Application Engine 106-2
Resource Usage Application Engine 106-2
118-1
118-2
100

FIG. 1

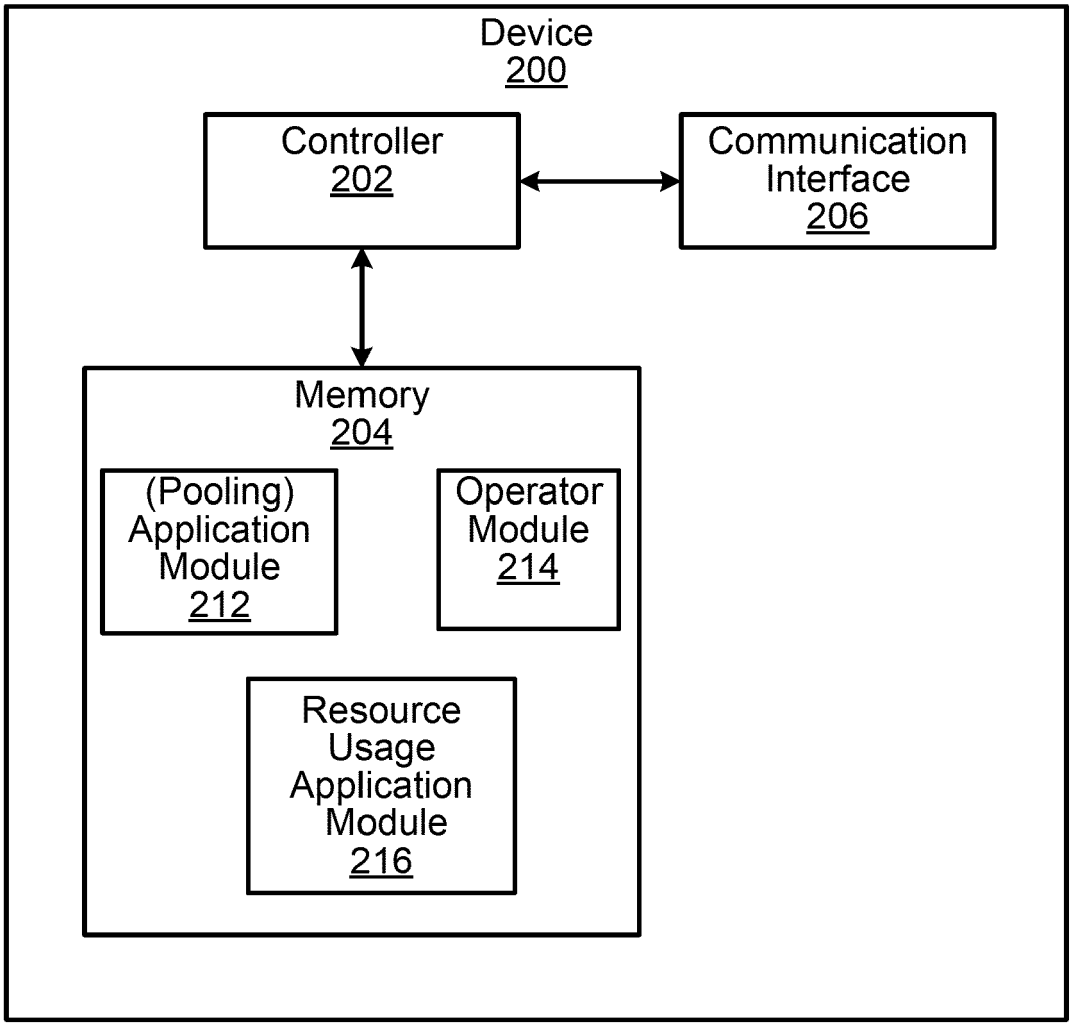
Device
200
Controller
202
Communication
Interface
206
Memory
204
(Pooling)
Application
Module
212
Operator
Module
214
Resource
Usage
Application
Module
216

FIG. 2

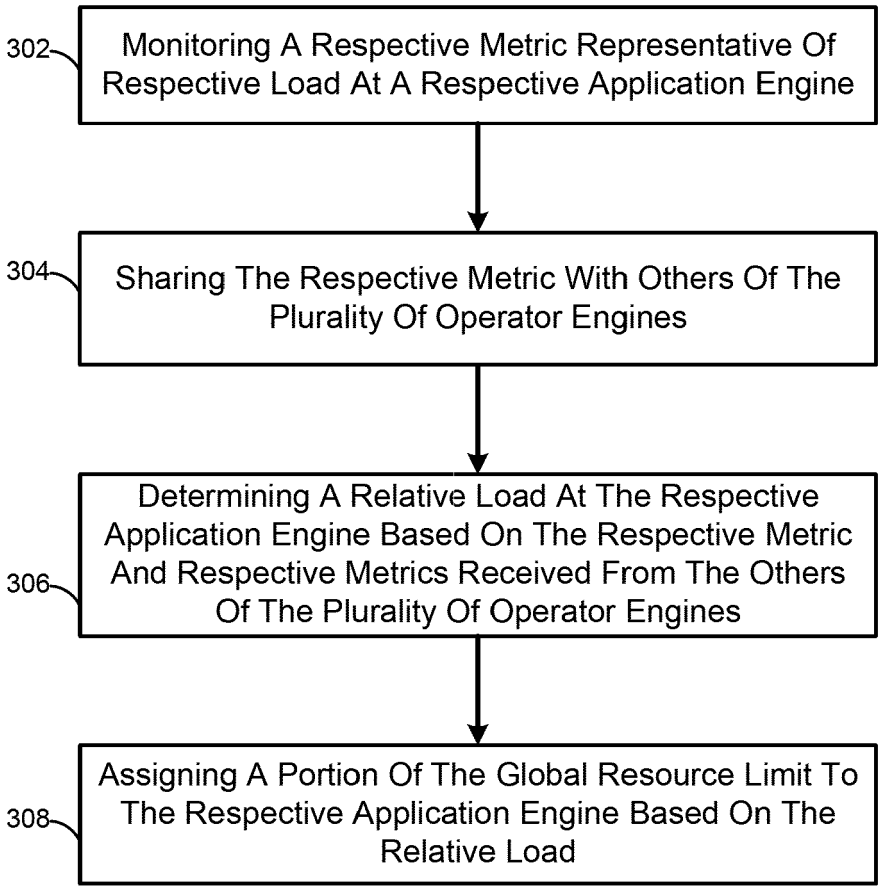
302
Monitoring A Respective Metric Representative Of Respective Load At A Respective Application Engine
304
Sharing The Respective Metric With Others Of The Plurality Of Operator Engines
306
Determining A Relative Load At The Respective Application Engine Based On The Respective Metric And Respective Metrics Received From The Others Of The Plurality Of Operator Engines
308
Assigning A Portion Of The Global Resource Limit To The Respective Application Engine Based On The Relative Load
300

FIG. 3

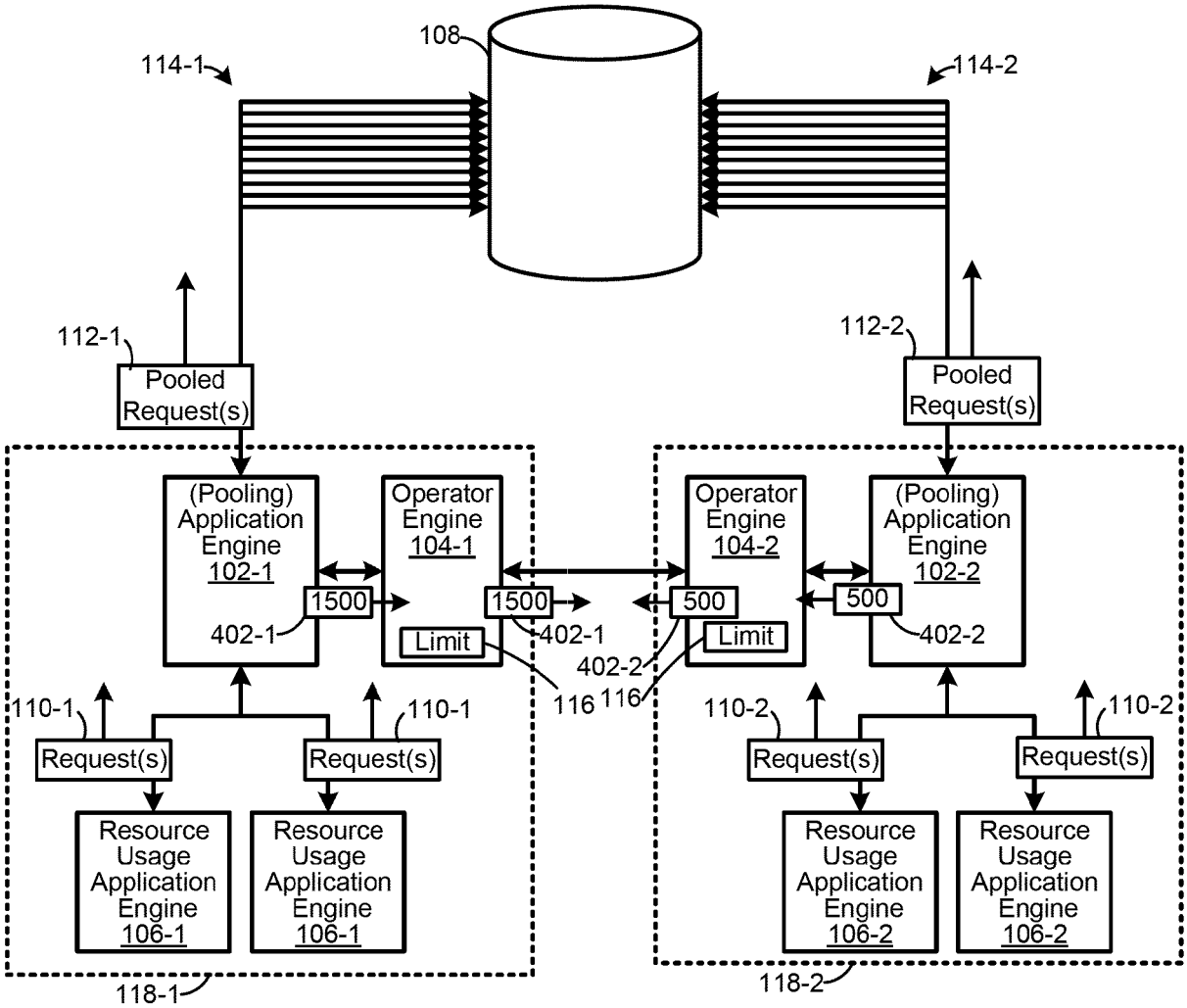
108
114-1
114-2
112-1
Pooled Request(s)
112-2
Pooled Request(s)
(Pooling) Application Engine 102-1
Operator Engine 104-1
Operator Engine 104-2
(Pooling) Application Engine 102-2
1500
1500
500
500
402-1
402-1
402-2
402-2
Limit
Limit
116
116
110-1
110-1
110-2
110-2
Request(s)
Request(s)
Request(s)
Request(s)
Resource Usage Application Engine 106-1
Resource Usage Application Engine 106-1
Resource Usage Application Engine 106-2
Resource Usage Application Engine 106-2
118-1
118-2
100

FIG. 4

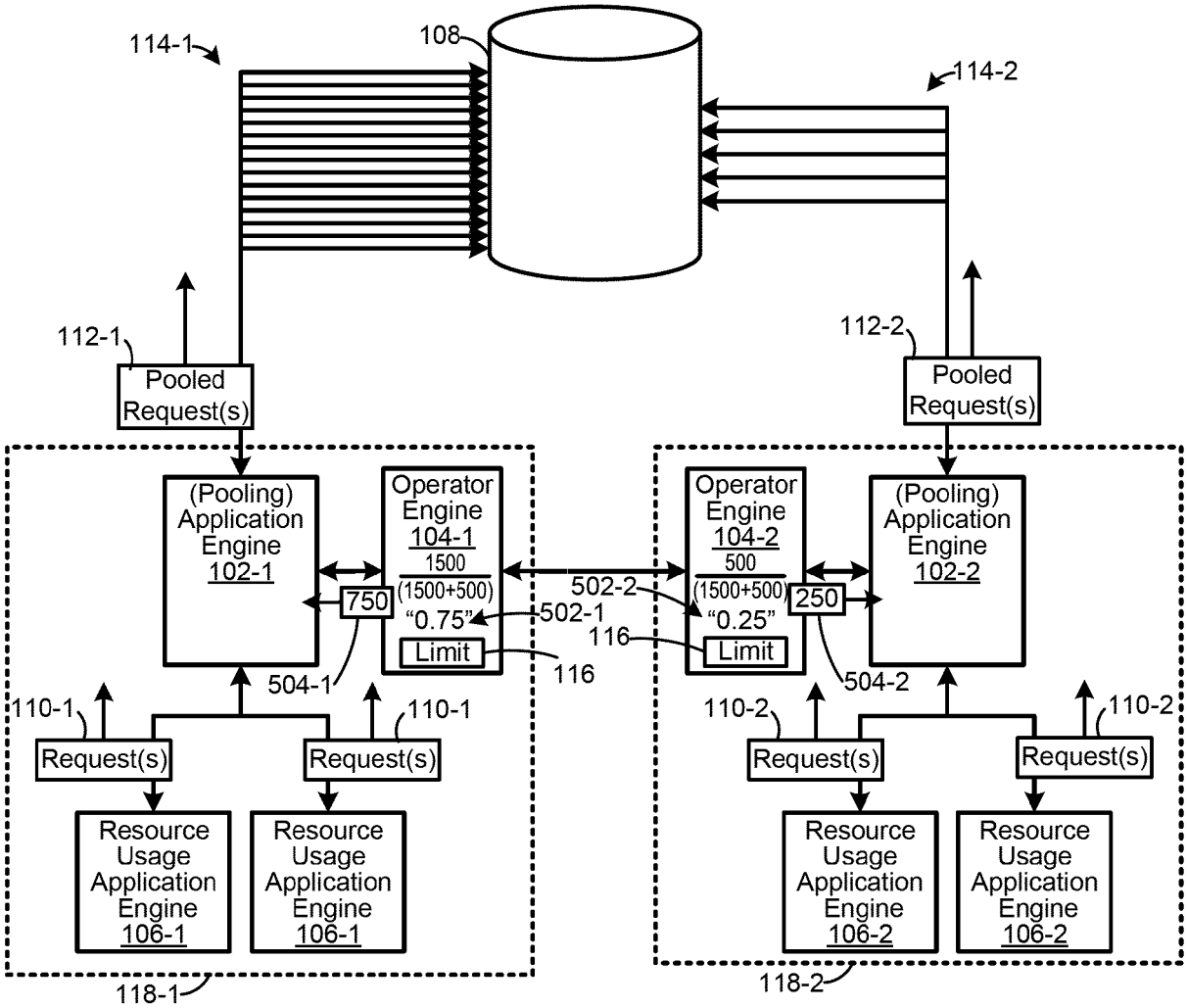
114-1
108
114-2
112-1
Pooled Request(s)
112-2
Pooled Request(s)
(Pooling) Application Engine 102-1
Operator Engine 104-1
1500
(1500+500)
"0.75"
750
Limit
116
502-1
502-2
116
Operator Engine 104-2
500
(1500+500)
"0.25"
250
Limit
(Pooling) Application Engine 102-2
504-1
504-2
110-1
Request(s)
110-1
Request(s)
110-2
Request(s)
110-2
Request(s)
Resource Usage Application Engine 106-1
Resource Usage Application Engine 106-1
Resource Usage Application Engine 106-2
Resource Usage Application Engine 106-2
118-1
118-2
100

FIG. 5

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 22 30 5139

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 2020/287962 A1 (MISHRA RAKESH [US] ET AL) 10 September 2020 (2020-09-10) * abstract * * paragraphs [0018] - [0021] * * paragraphs [0040] - [0096] *	1-15	INV. G06F9/50
A	US 2020/014609 A1 (HOCKETT HUGH EDWARD [US] ET AL) 9 January 2020 (2020-01-09) * the whole document *	1-15	

TECHNICAL FIELDS SEARCHED (IPC)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	11 June 2022	Renault, Sophie

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 22 30 5139

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

11-06-2022

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2020287962 A1	10-09-2020	NONE	
US 2020014609 A1	09-01-2020	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82