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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
03 December 2020 (03.12.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/243244 A1

(51) International Patent Classification:
G06F 12/00 (2006.01)

(21) International Application Number:
PCT/US2020/034835

(22) International Filing Date:
28 May 2020 (28.05.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/853,288 28 May 2019 (28.05.2019) US

(72) Inventor; and
(71) Applicant: RANKIN, John [US/US]; 8125 State Route 138, Williamsport, OH 43164 (US).

(74) Agent: SMITH, Adam, J. et al.; Standley Law Group LLP, 6300 Riverside Drive, Dublin, OH 43017 (US).

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

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

(54) Title: SUPPORTING A VIRTUAL MEMORY AREA AT A REMOTE COMPUTING MACHINE

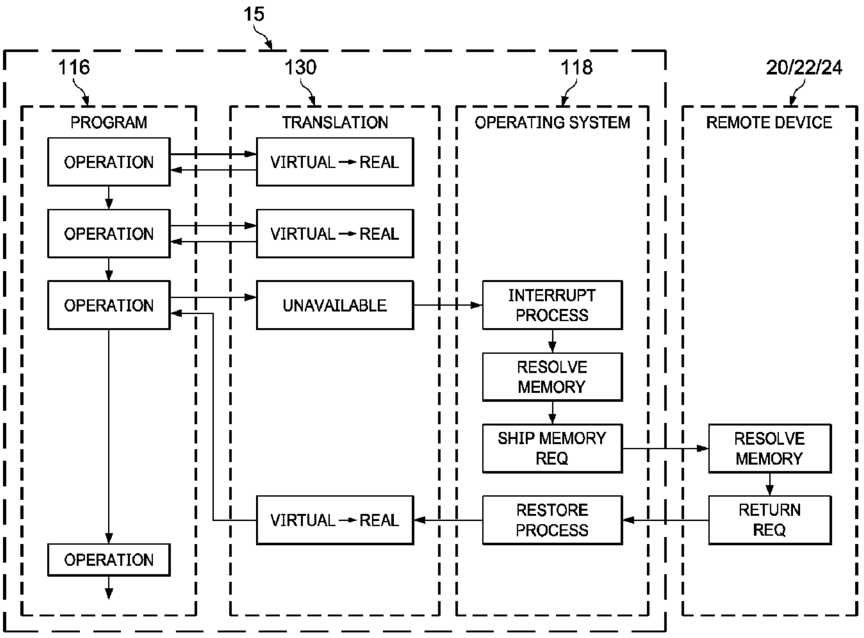

FIG. 4

(57) Abstract: Systems and methods for operating a virtual memory area (112) are provided. A dynamic address translation table (114) for the virtual memory area (112) is generated. A program is operated at a first computing machine (15) until insufficient local real memory (11) is available to complete operation. A request for real memory space is transmitted from the first computing machine (15) to an additional computing machine (20, 22, 24). A location of a segment of the local real memory (25, 27, 28) of the additional computing machine (20, 22, 24) is received at the first computing machine (20, 22, 24) and the dynamic address translation table (114) is updated to associate a virtual address with the received location.

[Continued on next page]

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

SUPPORTING A VIRTUAL MEMORY AREA AT A REMOTE COMPUTING MACHINE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent Application Serial No. 62/853,288 filed May 28, 2019, the disclosures of which are hereby incorporated by reference as if fully restated herein.

TECHNICAL FIELD

[0002] Exemplary embodiments relate generally to systems and methods for supporting virtual memory areas at remote computing machines.

BACKGROUND AND SUMMARY OF THE INVENTION

[0003] Computing machines that are designed to be used for general purpose computing needs generally have hardware configured to take advantage of the concept of virtual storage. Virtual memory has been in use since at least the 1960s and has allowed computing machines to overcome the limitations imposed by the space, size, and cost of real memory. Early computers were limited to executing and operating upon data that could completely fit within the memory storage that was available within a given machine. This storage area was often referred to as “real memory”, as it actually existed within the physical machine; however, this limitation significantly influenced the effectiveness of the computing machines and put additional limitations on the types of software programs that could be developed. To resolve this issue, hardware providers created a concept now often referred to as “virtual storage”. Virtual storage allows a software program to utilize more, and often far more, memory than is actually in existence within the machine. This is accomplished by mapping virtual to real addresses and taking advantage of the fact that not all memory is truly needed at all times, and therefore, sections of memory can be paged out or temporarily transferred to disk storage and returned to memory only when needed.

[0004] The fundamental idea behind virtual storage is the ability to appear to utilize far more memory space than is actually available. This is made possible because certain locations within memory are used less often than others. The concept is similar to reserves in banking, where the bank holds only the cash that is expected to be used rather than the entire aggregate that might be held in trust. It is unlikely that all customers will ask for all their money, all at the same time. Therefore, the bank needs to only have on hand what the bank reasonably expects will be requested during a given time period.

[0005] This internal hardware control over virtual storage is accomplished using two major controls. First, a translation table is maintained to facilitate mapping virtual addresses to real addresses. As an instruction is executed by the hardware processor, all addresses necessary for the machine instruction to be completed are resolved. This resolution process involves looking up the virtual address within the translation table and dynamically converting the virtual address to the corresponding real address. Once all addresses have been translated, the instruction can be executed and the illusion of the larger virtual area is complete.

[0006] Second, when the dynamic address translation process fails to produce a valid real address for use, an interruption of the normal machine program execution occurs. Control is passed to the machine’s operating system and gives the operating system the opportunity to manipulate memory. The operating system either assigns an available free piece of real memory that is added to the translation table, or swaps out a piece of lesser used real memory to make room for a piece of storage previously moved to long term storage. The dynamic translation table is updated and the interrupted program is resumed to permit the instruction execution to complete its normal operation.

[0007] Since the internal hardware of the computing machine is designed to allow for the use of virtual memory, these machines provide a wide range of support for multiple virtual memory areas within the machine. Therefore, a single operating program can have access to more than one virtual area at a time, or different programs within the machine can be assigned specific virtual memory areas for their operation.

[0008] The two major controls over virtual storage operation are tightly connected to both the hardware and the operating system of the given machine. The dynamic address translation table is managed by the operating system, along with the interruption method for switching between real memory usage and physical long-term storage. What is needed is a virtual memory system and method that is not tightly connected to the hardware of the computing machine or the operating system.

[0009] The present disclosures provide systems and methods where the internal dynamic address translation of a computing machine may be intercepted to permit programming processes to utilize virtual storage that exists outside of the computing machine's direct control. This may be accomplished by interfacing with the operating system of a computing machine and inserting a new communication process between the process interruption and the dynamic address translation. Rather than utilize only resources directly connected with the computing machine, the disclosed system and method may provide a virtual storage space located at, or shared with, another machine using a network connection. A local real memory and a local dynamic address translation table may serve to support the virtual storage located on a remote machine.

[0010] Further features and advantages of the systems and methods disclosed herein, as well as the structure and operation of various aspects of the present disclosure, are described in detail below with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] In addition to the features mentioned above, other aspects of the present invention will be readily apparent from the following descriptions of the drawings and exemplary embodiments, wherein like reference numerals across the several views refer to identical or equivalent features, and wherein:

[0012] **FIGURE 1** is a simplified diagram of a traditional virtual memory system;

[0013] **FIGURE 2** is a simplified flow diagram for executing a program while using virtual memory;

[0014] **FIGURE 3** is a simplified diagram of exemplary remote machines in accordance with the present invention;

[0015] **FIGURE 4** is a simplified flow diagram for executing a program using virtual memory located at, or shared with, the remote machines of figure 3 in accordance with the present invention; and

[0016] **FIGURE 5** is a diagram of an exemplary shared virtual area for use with the systems and processes of figures 3-4 in accordance with the present invention.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENT(S)

[0017] Various embodiments of the present invention will now be described in detail with reference to the accompanying drawings. In the following description, specific details such as detailed configuration and components are merely provided to assist the overall understanding of these embodiments of the present invention. Therefore, it should be apparent to those skilled in the art that various changes and modifications of the embodiments described herein can be made without departing from the scope and spirit of the present invention. In addition, descriptions of well-known functions and constructions are omitted for clarity and conciseness.

[0018] Embodiments of the invention are described herein with reference to illustrations of idealized embodiments (and intermediate structures) of the invention. As such, variations from the shapes of the illustrations as a result, for example, of manufacturing techniques and/or tolerances, are to be expected. Thus, embodiments of the invention should not be construed as limited to the particular shapes of regions illustrated herein but are to include deviations in shapes that result, for example, from manufacturing.

[0019] The traditional use of virtual memory is illustrated in **FIGURE 1**. In the illustrated example, 4M of real memory 10 is used to support and create 16M of virtual memory 12. When access to a piece of real memory 10 is required by a program 16, an address for the portion of real memory 10 corresponding with a portion of requested virtual memory 12 is retrieved from a translation table 14. The translation table 14 provides an entry for every segment of the virtual memory 12. By looking up the virtual memory address, the corresponding real memory address that is used to hold the data may be substituted by a computing machine. If the virtual memory 12 is not supported by an equivalent piece of real memory 10, the translation table 14 is updated. This updating process might require the temporary transfer of data to a more permanent location, like a computer disk or direct access storage device (DASD).

[0020] **FIGURE 2** is a simplified diagram illustrating a traditional process for executing a program 16 using virtual memory 12. As an exemplary illustration of how the traditional virtual memory system operates, the process is described with regards to an "add" operation. The "add" instruction may take a number stored in memory and mathematically add it to a register within the computing machine. This instruction may utilize virtual storage 12 and real memory 10 in two ways. First, the instruction itself may be located in virtual memory 12 and may be locatable by its virtual memory address. Second, the number to be added to the register may likewise be located in virtual memory 12 and may be separately locatable by its virtual memory address. These two virtual memory addresses are translated into their respective corresponding real memory addresses to determine where the instruction and data to be added is actually located in the real memory 10. This translation 30 may be performed by way of a translation table 14 stored at the hardware of the local machine. This translation 30 may be performed prior to the "add" operation executing. The translation 30 is performed prior to the instruction because the virtual memory 12 does not truly exist within the local machine. Instead, the virtual memory 12 is merely a convenient way of describing a contiguous area of storage that in reality is much larger than the local machine can actually contain in its local memory.

[0021] Each virtual address must undergo a translation to reveal its actual location within the real memory 10 that actually exists with the local machine. This translation is accomplished by the tightly coupled hardware through the use of the translation table 14.

The local machine takes the virtual address and looks it up at the translation table 14 to determine the corresponding real address. Once the translation of both addresses is completed, the operation may be executed. At no time is a program 116 involved aware of this translation as the translation is performed by the hardware of the local machine without knowledge of the program 116 (*i.e.,* the software).

[0022] However, if any of the requested virtual addresses do not have a valid representation within the translation table 14 the hardware must interrupt the instruction and pass control to an operating system 18 for the machine. Since this interruption occurs when insufficient real memory 10 exists to support the virtual image, the operating system 18 must resolve the problem. This resolution might involve connecting a free and available piece of real memory 10 or moving a previously stored piece of real memory 10 back from disk after making space available by storing a piece of real memory 10 that is no longer needed. Once the translation table 14 has been updated to reflect the now connected virtual to real relationship, the interrupted operation of the program 116 may be resumed. This process is highly confined and tightly coupled to the operation within the local hardware as well as the physical hardware of the local machine.

[0023] **FIGURE 3** is a simplified diagram of exemplary remote machines 100. **FIGURE 4** is a simplified flow diagram for executing a program while using virtual memory located at, or shared with, the remote machines 100. A local machine 15 may be in electronic communication with a remote machine 20. Such electronic communication may be made by way of an established communication pathway 19 between the local machine 15 and the remote machine 20. The communication pathway 19 may comprise any number of wired or wireless connections and various communication devices. The communication pathway 19 may comprise one or more networks 17 of the same or different type. The network 17 may be an internet, intranet, cellular network, the world wide web, LAN, some combination thereof, or the like. The local machine 15 may comprise a server, personal computer, tablet, smartphone, or other computing device. The remote machine 20 may comprise a server, personal computer, tablet, smartphone, or other computing device.

[0024] A dynamic address translation table 114 may be located at the remote machine 20. The dynamic address translation table 114 may be located at the real memory 28 of the remote machine 20. The local machine 15 may maintain a copy of the dynamic address translation table 114. The copy of the translation table 114 may be stored at the real memory 11 of the local machine 15. In exemplary embodiments, the local machine 15 may, from time to time, utilize its copy of the dynamic address translation table 114 when the program 116 requests data stored at a virtual memory area 112 to translate the requested virtual address to the corresponding real address. However, where the local machine 15 is unable to resolve a request for translation, the request may be transmitted to the remote machine 20 for resolution at the dynamic address translation table 114 stored at the remote machine 20. The remote machine 20 may transmit an address for its real memory 28 to the local machine 15. The local machine 15 may use the transmitted address for the local machine’s image of the virtual memory area 112. The local machine 15 may update its copy of the translation table 114 accordingly.

[0025] The virtual memory area 112 may be contained entirely within the remote machine 20. In this way, a pseudo process is not necessarily required. Furthermore, implementation of the virtual to real translation may be made by way of the local machine 15.

[0026] In exemplary embodiments, the local machine 15 and/or the remote machine 20 may be in electronic communication with one or more additional member machines 22 and 24. Each of the additional member machines 22, 24 may have their own respective real memory 25, 27. The additional member machines 22 and 24 may each comprise a server, personal computer, tablet, smartphone, or other computing device. The electrical connection to the member machine 22 and 24 may be made by way of the same or a different network 17 along various established communication pathways 19, 21, and 23. Any number of additional member machines 22, 24 of any type may be utilized.

[0027] **FIGURE 5** is a diagram of an exemplary shared virtual area 112. In exemplary embodiments, without limitation, the virtual memory area 112 may be shared among the additional member machines 22, 24. The virtual memory area 112 may be local to one machine 15 and remote to the other member machines 22, 24. In this way, the local machine 15 as well as the additional member machines 22, 24 may utilize the same virtual memory area 112 such that the virtual memory area 112 may be pooled between the various member machines 15, 22, 24. The master copy of the translation table 114 may be located at the remote machine 20 such that requests for resolution of address issues may be resolved by the remote machine 20. Furthermore, the real memory 28 of the remote machine 20 may be copied and shared with the various member machines 15, 22, 24.

[0028] Turing back to an example regarding the “add” operation, an explanation is provided as to how this operation may be performed with the systems and methods of Figures 3-4. Of course, the “add” operation is merely exemplary, any type or kind of operation may be utilized in conjunction with any type or kind of program 116. When the virtual addresses for each of the required data segments are translated by the translation table 114, the resulting real address may point to a secondary or temporary container of the data. This is because the virtual space 112 being translated does not exist within the local machine 15 and may instead be a reference to a virtual memory area 112 which exists in the remote machine 20. A translation process 130, while managed by the hardware of the local machine 15, does not make reference to a tightly coupled virtual to real relationship.

[0029] When the dynamic translation process 130 reaches a point where the dynamic address translation table 114 indicates an invalid relationship, the memory resolution process performed by the operating system 118 may become more complex. Rather than dealing with local real memory 11 and local memory (e.g., disk) storage of the local machine 15, the operating system 118 may transmit its request for real memory to the remote machine 20. The remote machine 20 may perform the dynamic virtual to real address translation 130 by way of the translation table 114 located at the remote machine 20. The remote machine 20 may isolate an appropriate portion of the real memory 28 of the remote machine 20 and transmit a copy of this real memory 28 segment to the local machine 15. After the local machine 15 receives a copy of the remote real memory 28, the local machine 15 may populate an appropriate portion of its real memory 10 in a way that matches the remote real memory 28 segment. The local machine 15 may update its copy of the dynamic address translation table 114. The operation may resume its execution and utilize the remote real memory 28 in a virtual capacity as if it were local to the local machine 15. In this way, a local program 116 may operate upon data that exists in the remote machine 20. Furthermore, this operation may be transparent to the program 116. While hardware of the local machine 15 may be used for the translation 130, the virtual storage area 112 may be provided by the remote machine 20.

[0030] In exemplary embodiments, the same or a similar process may be utilized with respect to the other member machine 22 and 24. The translation table 114 may be stored at one of the machines 15, 22, 24, another remotely located, but electronically connected computing machine, some combination thereof, or the like.

[0031] Any embodiment of the present invention may include any of the features of the other embodiments of the present invention. The exemplary embodiments herein disclosed are not intended to be exhaustive or to unnecessarily limit the scope of the invention. The exemplary embodiments were chosen and described in order to explain the principles of the present invention so that others skilled in the art may practice the invention. Having shown and described exemplary embodiments of the present invention, those skilled in the art will realize that many variations and modifications may be made to the described invention. Many of those variations and modifications will provide the same result and fall within the spirit of the claimed invention. It is the intention, therefore, to limit the invention only as indicated by the scope of the claims.

[0032] Certain operations described herein may be performed by one or more electronic devices. Each electronic device may comprise one or more processors, electronic storage devices, executable software instructions, and the like configured to perform the operations described herein. The electronic devices may be general purpose computers or specialized computing devices. The electronic devices may be personal computers, smartphones, tablets, databases, servers, or the like. The electronic connections described herein may be accomplished by wired or wireless means. The computerized hardware, software, components, systems, steps, methods, and/or processes described herein may serve to improve the speed of the computerized hardware, software, systems, steps, methods, and/or processes described herein.

CLAIMS

What is claimed is:

1. A method for operating a virtual memory area shared between multiple, remotely located computing machines, said method comprising the steps of:

providing a number of computing machines, wherein each of said number of computing machines are located remote from one another, wherein each of said number of computing machines are in electronic communication with one another by way of one or more networks, and wherein each of said number of computing machines comprise local real memory;

generating a dynamic address translation table for the virtual memory area;

operating a program at a first one of the number of computing machines ("the first computing machine") requiring memory space;

determining, at the first computing machine, that insufficient local real memory is available to complete operation of the program;

transmitting a request for real memory space from the first computing machine to at least one other one of the number of computing machines ("the additional computing machine");

receiving, at the first computing machine from the additional computing machine, a location of a segment of the local real memory of the additional computing machine; and

updating the dynamic address translation table to associate a virtual address with the received location.

2. The method of claim 1 further comprising the steps of:

resuming operation of the program.

3. The method of claim 2 wherein:

the request for real memory space defines an amount of memory required to complete operation of the program; and

the segment of the local real memory is at least as large as the amount of memory required to complete operation of the program.

4. The method of claim 3 wherein:

operation of the program is halted until the dynamic address translation table is updated to associate the received location with the virtual address associated with the request for real memory space.

5. The method of claim 1 further comprising the steps of:

receiving, at the first computing machine, a request for real memory space from the additional computing machine;

transmitting, from the first computing machine to the additional computing machines, a location of a segment of the local real memory of the first computing machine; and

updating the dynamic address translation table to associate the transmitted location with a second virtual address.

6. The method of claim 1 wherein:

   the dynamic address translation table is stored at the first computing machine.

7. The method of claim 1 wherein:

   at least one of the number of computing machines comprises a personal computer.

8. The method of claim 1 wherein:

   at least one of the number of computing machines comprises a server.

9. The method of claim 1 wherein:

   at least one of the number of computing machines comprises a tablet.

10. The method of claim 1 wherein:

    said one or more networks comprise IP protocol networks.

11. A method for operating a virtual memory area shared between multiple, remote computing machines, said method comprising the steps of:

    establishing a connection between a local computing machine and a remote computing machine;

    operating a program at the local computing machine;

    defining, at the local computing machine, a remote virtual memory area;

    creating, at the local computing machine, a local dynamic address translation table to manage a local image of the remote virtual memory area;

transmitting a request for real memory from the local computing machine to the remote computing machine;

resolving virtual to real memory addresses at the remote machine by way of a remote dynamic address translation table stored at the remote computing machine and selection of an appropriate piece of real memory of the remote computing machine that satisfies the request for real memory;

transmitting a copy of the appropriate piece of real memory of the remote computing machine to the local computing machine;

updating the local dynamic address translation table at the local computing machine to allow the use of the transmitted copy of the appropriate piece of real memory; and

restoring operation of the program at the local computing machine.

12. The method of claim 11, further comprising the steps of:

establishing additional connections between the local computing machine and additional remote computing machines, wherein each of said additional remote computing machines comprise a respective dynamic address translation table.

13. The method of claim 12, further comprising the steps of:

synchronizing the local dynamic address translation table, each of the respective dynamic address tables, and the remote dynamic address translation table.

14. A system for operating a virtual memory area comprising:

a first computing machine comprising local memory;

a second computing machine located remote from the first computing machine and comprising local memory, wherein the second computing machine is in electronic communication with the first computing machine by way of a network;

a dynamic address translation table for the virtual memory area stored at the first computing machine; and

software instructions stored at the local memory of the first computing machine, which when executed, configures the first computing machine to:

operate a program until a determination is made that insufficient local memory is available to complete operations of the program;

transmit a request to the second computing machine for a segment of memory at least as large as an amount of memory required to complete operation of the program;

receive a location of a segment of the local memory of the second computing machine; and

update the dynamic address translation table by associating the received location with a virtual address.

15. The system of claim 14 further comprising:

additional software instructions stored at the local memory of the first computing machine, which when executed, configures the first computing machine to:

pause operations of the program until the dynamic address translation table is updated; and resuming operations of the program after the dynamic address translation table is updated.

16. The system of claim 14 further comprising:

a second dynamic address translation table for the virtual memory area stored at the second computing machine.

17. The system of claim 16 further comprising:

software instructions stored at the local memory of the second computing machine, which when executed, configures the second computing machine to update the second dynamic address translation table to reflect the dynamic address translation table.

18. The system of claim 16 further comprising:

software instructions stored at the local memory of the second computing machine, which when executed, configures the second computing machine to update the second dynamic address translation table by associating the transmitted location with the virtual address.

19. The system of claim 14 wherein:

the network is a packet switched network.

20. The system of claim 19 further comprising:

a number of additional computing machines, each comprising local real memory comprising a respective dynamic address translation table and, wherein each of said number of additional computing devices are located remote from said first and second computing devices and are connected to said first and second computing devices by way of said network, wherein said first computing machine, said second computing machine, and each of said additional computing machine are configured to synchronize their respective dynamic address translation tables.

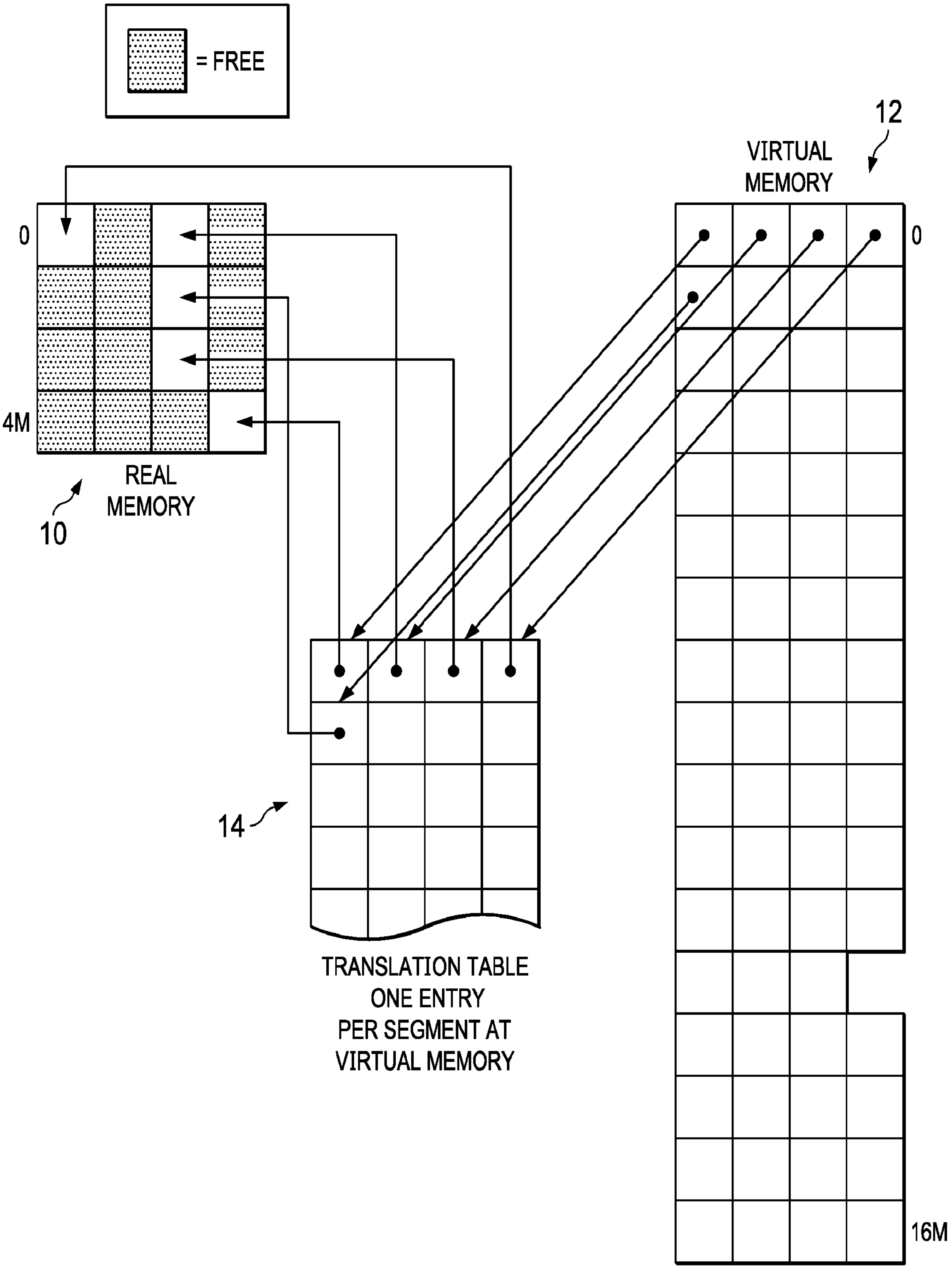

FIG. 1
(PRIOR ART)

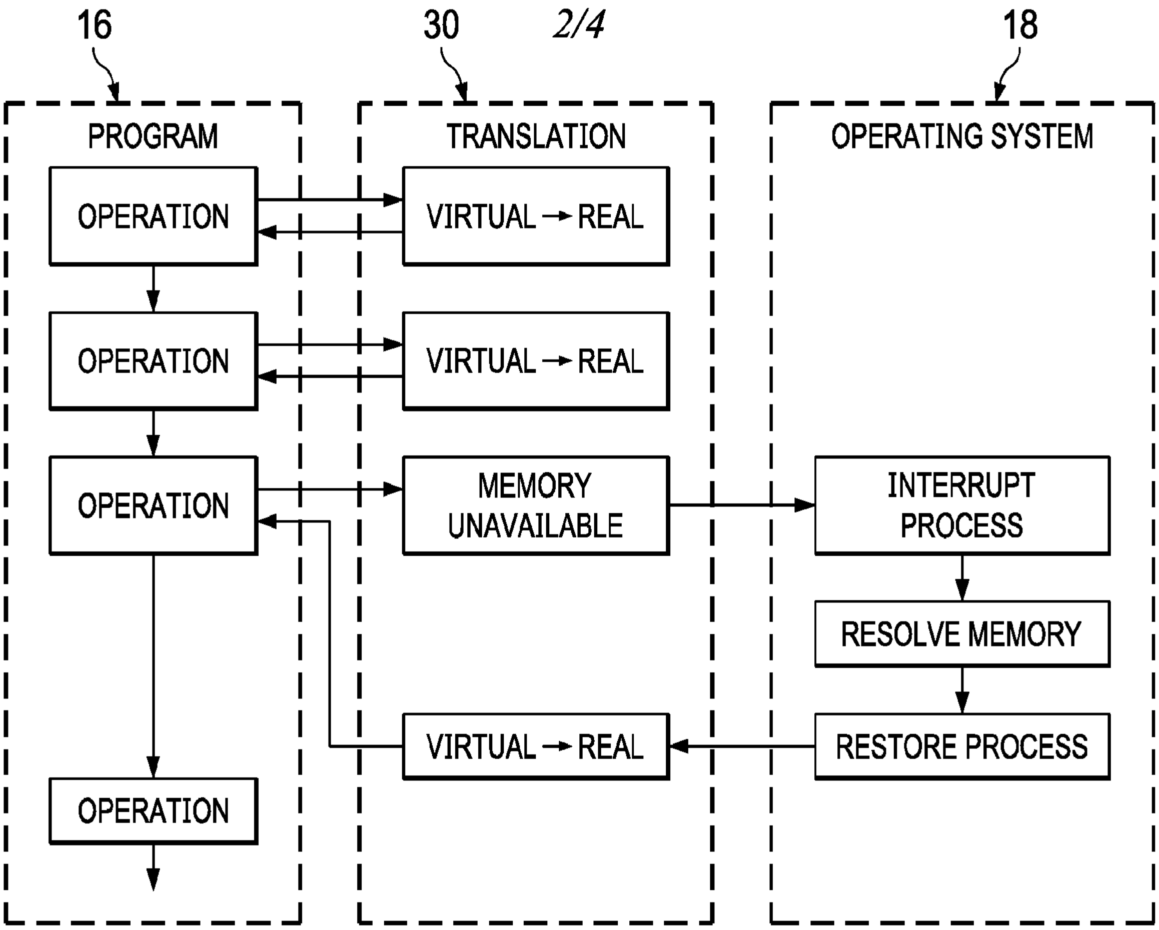

FIG. 2
(PRIOR ART)

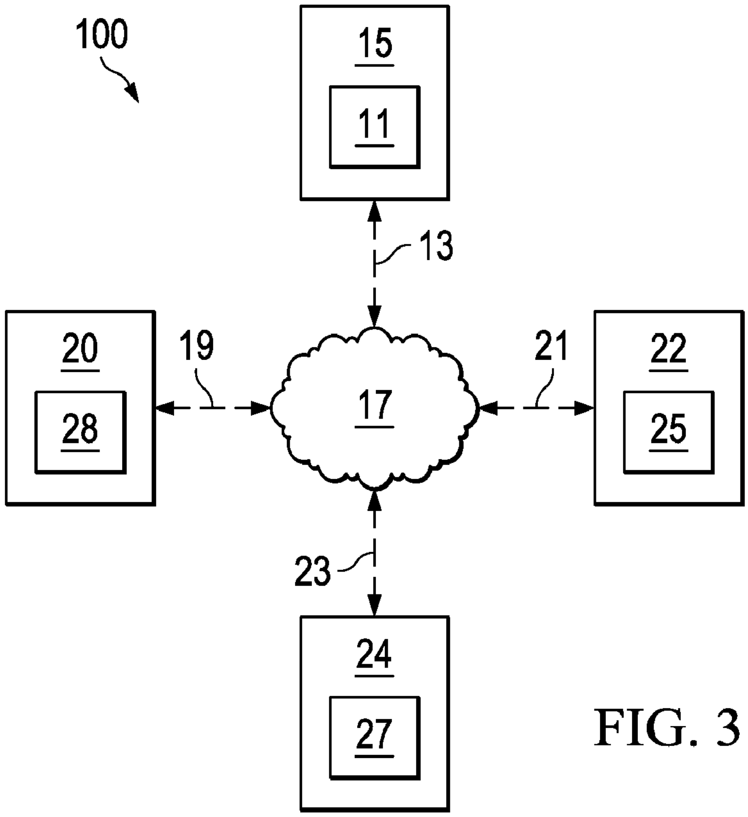

FIG. 3

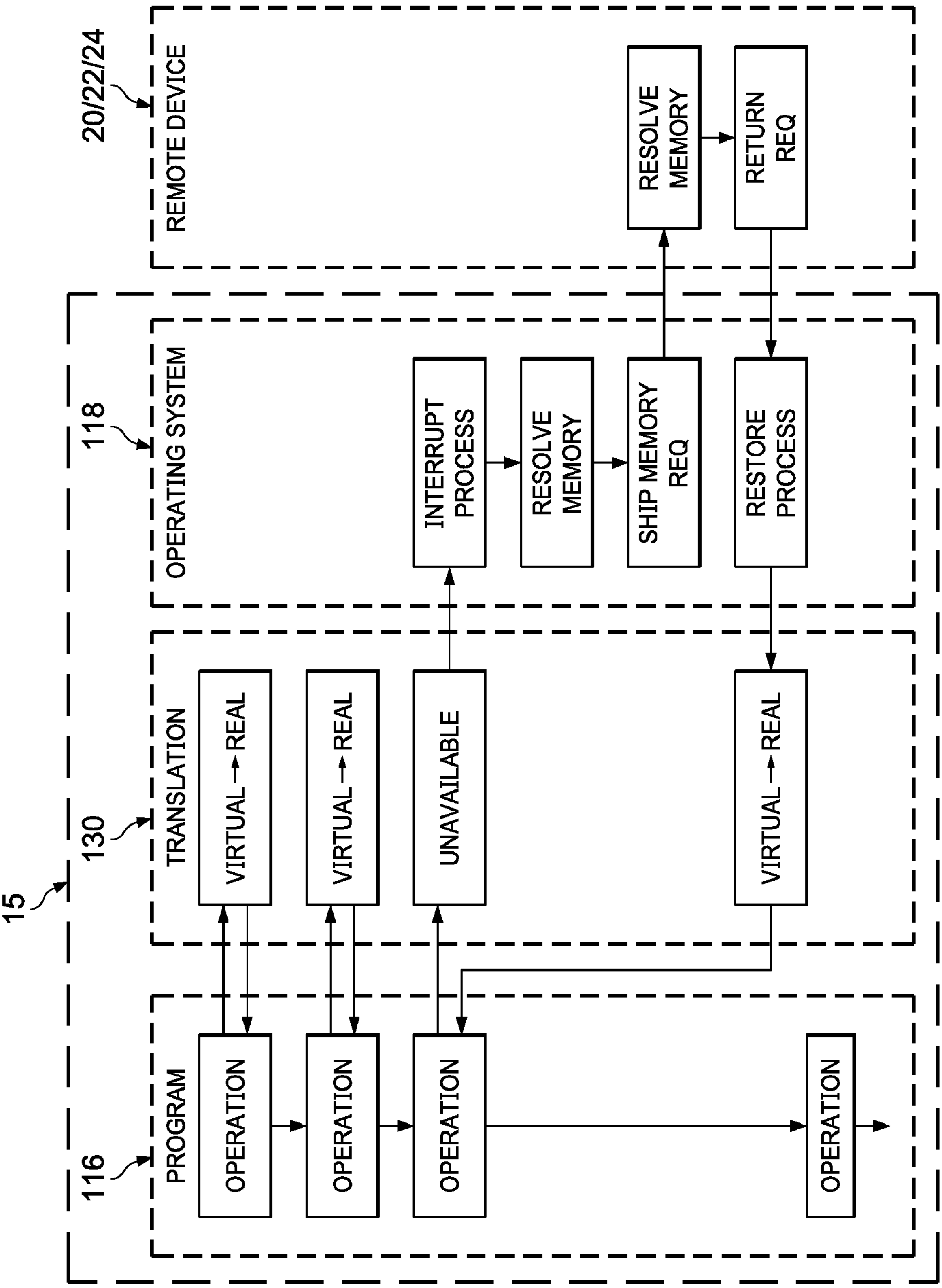
15
116
PROGRAM
OPERATION
OPERATION
OPERATION
OPERATION
130
TRANSLATION
VIRTUAL→REAL
VIRTUAL→REAL
UNAVAILABLE
VIRTUAL→REAL
118
OPERATING SYSTEM
INTERRUPT PROCESS
RESOLVE MEMORY
SHIP MEMORY REQ
RESTORE PROCESS
20/22/24
REMOTE DEVICE
RESOLVE MEMORY
RETURN REQ

FIG. 4

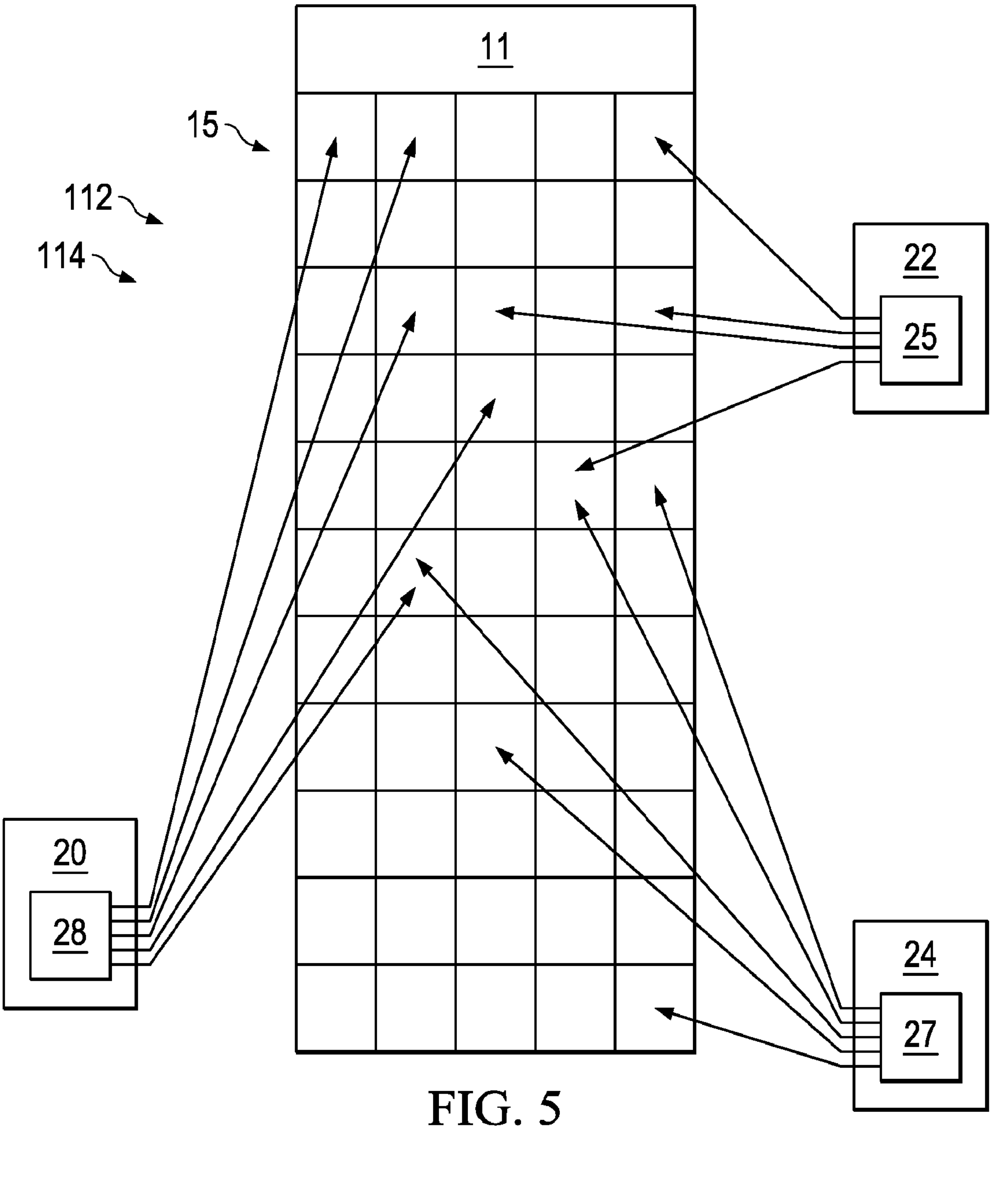

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 20/34835

A. CLASSIFICATION OF SUBJECT MATTER

IPC - G06F 12/00 (2020.01)

CPC - G06F 12/00, G06F 12/02

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2014/0280669 A1 (Microsoft Corporation) 18 September 2014 (18.09.2014); entire document, esp. [0005], [0008], [0017]-[0018], [0020], [0024], [0030], [0037]-[0038], Fig. 1-2	1-20
A	US 2011/0072234 A1 (Chinya et al.) 24 March 2011 (24.03.2011); entire document	1-20

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

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

"D" document cited by the applicant in the international application

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

14 August 2020 (14.08.2020)

Date of mailing of the international search report

11 SEP 2020

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-8300

Authorized officer
Lee Young

Telephone No. PCT Helpdesk: 571-272-4300