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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 September 2010 (02.09.2010)

PCT

(10) International Publication Number
WO 2010/097306 A1

(51) International Patent Classification:
G06F 1/32 (2006.01)

(21) International Application Number:
PCT/EP2010/051856

(22) International Filing Date:
15 February 2010 (15.02.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/393,475 26 February 2009 (26.02.2009) US

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(71) Applicant *(for MG only)*: **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, North Harbour, Portsmouth, Hampshire PO6 3AU (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BOZEK, James** [US/US]; IBM Corporation, Dept. XYSA/Bldg. 002, PO Box 12195, 3039 Cornwallis Road, Research Triangle Park, NC 27709-2195 (US). **SUFFERN, Edward** [US/US]; IBM Corporation, MP C8RA/205, 3039 E Cornwallis Road, PO Box 12195, Research Triangle Park, NC 27709-2195 (US). **WOOLDRIDGE, James** [US/US]; IBM Corporation, 3600 Carillon Point, Kirkland, WA 98033-7304 (US).

(74) Agent: **STRETTON, Peter, John**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester, Hampshire SO21 2JN (GB).

(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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

(54) **Title**: POWER MANAGEMENT TO MAXIMIZE REDUCED POWER STATE FOR VIRTUAL MACHINE PLATFORMS

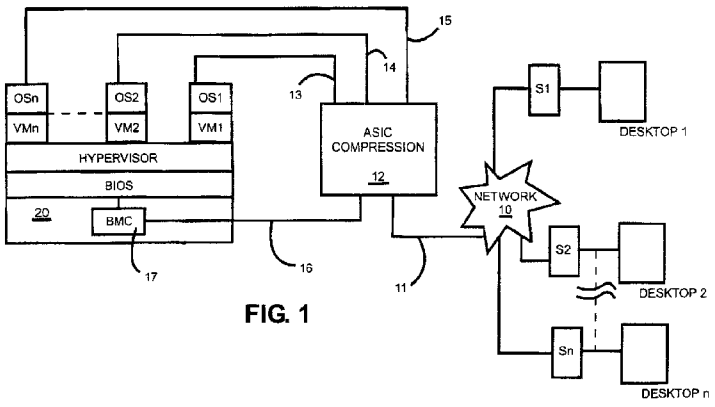

FIG. 1

(57) **Abstract**: Power management in a virtual machine environment that includes at least one virtual machine platform providing a plurality of virtual machines, and a plurality of separate (user) computers, each connected to a respective one of the virtual machines in a typical virtual machine distribution environment. There is also provided a function, independent of the connections of the user computers to the virtual machines, for determining if each of said computers connected to the virtual machines is in an active state together with a function for switching the virtual machine platform into a reduced power consumption state in the platform when all of the computers connected to virtual machines are in a non-active state.

WO 2010/097306 A1

POWER MANAGEMENT TO MAXIMIZE REDUCED POWER STATE FOR VIRTUAL MACHINE PLATFORMS

Technical Field

The present invention relates to virtual machine platforms including a plurality of virtual machines, each respectively connected to each of a plurality of separate computers and particularly to power management for such virtual machine platforms.

Background of Related Art

In the past ten years, with the great increase in World Wide Web (Web) systems, the computer processing power required by an organization has grown exponentially each year so that now hundreds and even thousands of servers are required. This has led to a resurgence of larger and larger mainframe computers. Particularly mainframe and like large computers operating in a virtual machine (VM) mode in which multiple instances of an operating system and associated application program reside in the same physical hardware. Such virtual machines have been satisfying the needs for the large number of servers that are often arrayed as virtual machine server farms. For further background, attention is directed to the article: Virtual Linux servers under z/VM: security, performance, and administrative issues, D. Turk, published in the IBM Systems Journal, July 2005; and to the article: More POWER to Ya, Expanded Virtualization Manager capabilities help customers grow and manage virtualized environments, Jim Fall, published in the IBM Systems Magazine, September 2007.

In such virtual machine environments wherein multiple user computers are connected to each virtual machine platform providing a plurality of virtual machines respectively connected to these multiple users, power management is difficult to control. Full power is required at each virtual machine platform prior to the initiation of a virtual machine session. Since the virtual machine platform must always be available to remote user computers that need to access appropriate virtual machines, it has been customary to continuously operate any online platform in a full power mode. The wasted power consumption becomes

particularly pronounced when the virtual machine platforms are arrayed as virtual machine server farms.

Summary of the Present Invention

The present invention accordingly provides, in a first aspect, a system for power management in a virtual machine environment comprising: a virtual machine platform providing a plurality of virtual machines; a plurality of separate client devices, each connected to a respective one of said virtual machines; means, independent of said connections of said computers to said virtual machines, for determining if each of said client devices connected to said virtual machines is in an active state; and means for switching said virtual machine platform into a reduced power consumption state when all of said client devices connected to virtual machines are in a non-active state.

Preferably, said plurality of client devices are remote from said virtual machine platform; and further including network means for respectively connecting said plurality of client devices to said virtual machine platform. Preferably, said network means are World Wide Web network means. The system preferably further includes a plurality of said virtual machine platforms, each platform respectively providing a plurality of virtual machines; and

a plurality of sets, each set including a plurality of client devices respectively connected to the virtual machines in one of said plurality of virtual machine platforms; wherein said means for switching switch a respective one of said virtual machine platforms into a reduced power consumption state when all of said client devices connected to virtual machines in said respective one platform are in a non-active state.

Preferably, said plurality of said virtual machine platforms are servers arranged as a virtual server farm; and each of said plurality of sets of client devices are client computers connected to the virtual machines in the respective servers. Preferably, said means for switching said virtual machine platform into said reduced power state are in said virtual machine platform; and further including means for communicating whether each of said client devices are in a non-active state to said means for switching by a path independent of an operating system of said virtual machine platform. Preferably, said virtual machine

platform includes a hypervisor and a BIOS; and said path for communicating the non-active state is connected to the BIOS independent of said hypervisor. Preferably, said virtual machine platform includes a baseboard management controller (BMC) having means for tracking the non-active states of said client devices; and said path for communicating the non-active state is connected to the BIOS through said BMC.

In a second aspect, there is provided a method for power management in a virtual machine platform including a plurality of virtual machines, each respectively connected to each of a plurality of separate client devices comprising: determining, independently of said connections of said client devices to said virtual machines, if each of said client devices connected to said virtual machines is in an active state; and switching said virtual machine platform into a reduced power consumption state when all of said client devices connected to virtual machines are in a non-active state.

Preferably, said plurality of client devices are remote from said virtual machine platform; and respectively connecting said plurality of client devices to said virtual machine platform through a network. Preferably, said network is the World Wide Web. Preferably, said power management is for a plurality of said virtual machine platforms, each platform respectively providing a plurality of virtual machines and includes: connecting a plurality of sets, each set including a plurality of client devices, respectively to the virtual machines in one of said plurality of virtual machine platforms; wherein a respective one of said virtual machine platforms is switched into a reduced power consumption state when all of said client devices connected to virtual machines in said respective one platform are in a non-active state. Preferably, said switching of said virtual machine platform into said reduced power state is controlled in said virtual machine platform; and further including the step of communicating whether each of said client devices are in a non-active state to said virtual machine platform by a path independent of an operating system of said virtual machine platform Preferably, said path for communicating the non-active state is connected to the BIOS of the virtual machine platform independent of the hypervisor of the virtual machine platform.

In a third aspect, there is provided a computer program comprising computer program code to, when loaded into a computer system and executed thereon, cause said computer to perform all the steps of the method of the second aspect.

The computer program of the third aspect may be embodied in the form of a computer usable medium having stored thereon a computer readable program for power management in a virtual machine platform including a plurality of virtual machines, each respectively connected to each of a plurality of separate client devices, wherein the computer readable application program when executed on a computer causes the executing computer to: determine, independently of said connections of said client devices to said virtual machines, if each of said client devices connected to said virtual machines is in an active state; and switch said virtual machine platform into a reduced power consumption state when all of said client devices connected to virtual machines are in a non-active state.

Preferably, said plurality of client devices are remote from said virtual machine platform; and said plurality of client devices are respectively connected to said virtual machine platform through a network. Preferably, said network is the World Wide Web. Preferably, said power management is for a plurality of said virtual machine platforms, each platform respectively providing a plurality of virtual machines and said computer program when executed connects a plurality of sets, each set including a plurality of client devices, respectively to the virtual machines in one of said plurality of virtual machine platforms; and

switches a respective one of said virtual machine platforms into a reduced power consumption state when all of said client devices connected to virtual machines in said respective one platform are in a non-active state.

Preferably, said computer program when executed causes the executing computer to: control the switching of said virtual machine platform into said reduced power state within said virtual machine platform; and communicate whether each of said client devices are in a non-active state to said virtual machine platform by a path independent of an operating system of said virtual machine platform. Preferably, said computer program when executed causes the executing computer to communicate the non-active state to the BIOS of the virtual machine platform independent of the hypervisor of the virtual machine platform.

The present invention thus addresses the power consumption problem of maintaining each virtual machine platform in a full power mode even when there are no user computers connected to the virtual machine platform in an active mode.

To this end, embodiments of the present invention provide a system, method and computer program for power management in a virtual machine environment that includes at least one physical virtual machine platform providing a plurality of virtual machines, and a plurality of separate (user) client devices each connected to a respective one of the virtual machines in a typical virtual machine distribution environment. Client devices are understood to include user computers and computer subsystems, including printers, disk drives and serial ports, among others. In the description of the following invention, when the term user computer is used, it is intended to include all such client devices.

There is also provided a function, independent of the connections of the user computers or client devices to the virtual machines, for determining if each of said client devices connected to the virtual machines is in an active state together with a function for switching the virtual machine platform into a reduced power consumption state in the platform when all of the user computers connected to the virtual machines are in a non-active state.

In preferred operations, the implementation of this invention is when the plurality of user computers are remote from the virtual machine platform, and there are networks, usually the Internet or Web (these two terms are used interchangeably in this description) for respectively connecting the plurality of user computers to the virtual machine platform.

As set forth above, the preferred embodiment of the present invention is particularly advantageous in systems, e.g. virtual server farms, wherein there are a plurality of the virtual machine platforms, each platform respectively providing a plurality of virtual machines, together with a plurality of user sets, each user set including a plurality of user computers or devices respectively connected to the virtual machines in one of the plurality of virtual machine platforms, so that the means for switching switch a respective one of the virtual machine platforms into a reduced power consumption state when all of the user computers or client devices connected to virtual machines in the respective one platform are in a non-

active state. Such an implementation may be effectively used when the plurality of the virtual machine platforms provide servers arranged as a virtual server farm, and each of the plurality of sets of user computers are client devices or user computers connected to the virtual machines in the respective servers.

As will be hereinafter set forth in greater detail, in a preferred embodiment of this invention, the implementation for switching the virtual machine platform into the reduced power state is in the virtual machine platform and the means for communicating whether each of said computers are in a non-active state are connected to the means for switching by a path independent of an operating system of said virtual machine platform. This path for communicating the non-active state is connected to the Basic Input Output System (BIOS) firmware of the virtual machine platform by a route independent of said hypervisor of the platform. In such an arrangement, the virtual machine platform includes a baseboard management controller (BMC) having a function for tracking the non-active states of the connected user computers and the path for communicating the non-active state is connected to the BIOS through the BMC.

Brief Description of the Drawings

A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Fig. 1 is a generalized diagrammatic view of a network portion, including a single representative virtual machine platform and a set of remote user computers or client devices connected to virtual machines in the platform to illustrate the non-active state monitoring of the user computers;

Fig. 2 is a diagrammatic view like that of Fig. 1 showing a plurality of virtual machine platforms respectively connected to a plurality of sets of user computers;

Fig. 3 is a general flowchart of a program set up to implement a preferred embodiment of the present invention for power management in a virtual machine environment; and

Fig. 4 is a flowchart of an illustrative run of the program set up in Fig. 3.

Detailed Description of the Preferred Embodiment

Referring to Fig. 1, there is shown a generalized diagrammatic view of a network portion, including a single representative virtual machine platform 20, and a set of remote user computers, i.e. client devices, Desktops 1, 2 and n connected via network 10, e.g. the Web, and respective servers S1, S2 and Sn, to virtual machines, VM1, VM2 and VMn in the platform, using operating systems OS1, OS2 and OSn.

The assignment of specific virtual machines, VMs to specific user computers, desktops 1 through n, is allocated by hypervisor 21, the virtual machine platform supervisor. The hypervisor manages the multiple operating systems, OS1 - OSn, as well as the platform's processor, BIOS 22, memory and other resources as required. Since this is a virtual machine platform, the hypervisor may be considered to be a virtualization manager. Current conventional VM connection brokers (not shown) operate to assign the currently running virtual machines on platform 20 to specific client devices illustrated by user desktops 1 to n. However, such connection brokers are only involved in the login and disconnect processes. Thus, if a particular desktop 1, 2 or n logs in and then is idle, the particular assigned VM and its associated resources in the virtual machine platform 20 remain fully powered irrespective of the non-activity of assigned user desktop 1 through n computers.

Accordingly, the present invention provides an implementation independent from the operating systems and the hypervisor 21 of the platform 20 to detect non-activity of the allocated user desktop 1-n computers. Thus, in the environment of Fig. 1, the hypervisor, which may be commercially available in a VMware, Citrix or Calista environment, has an individual software component working with hardware compression engines that allow the multiple users (desktops 1-n) to access the same platform 20. Thus, when the platform hardware is powered on, the hypervisor allocates memory resources specific for that user. When the user no longer requires the resources, e.g. logs off, the resources are released back into the platform pool. However, the platform 20 remains fully powered on and fully functional, even though the platform is no longer required.

The present invention, as will be hereinafter described in greater detail, provides for putting the platform into a lower power consumption state when virtual machine resources (processor, memory, operating system and application software) are no longer required by the user computers o client devices, i.e. all are in the non-active state; but when a user computer or client device does require the use of a VM supported by the platform, i.e. the user computer becomes active, the platform 20 exits the low power state and becomes fully operational.

The implementation of the present invention takes advantage of existing apparatus for changing power states of a virtual machine platform 20. The BIOS 22 provides industry standard Advanced Configuration and Power Interface (ACPI) states that are able to remove power from different parts of the platform 20.

In the implementation of the invention, there is a path through network 10 (may be the Internet or Web) via connection 11 through ASIC (Application-Specific Integrated Circuit) compression core 12 wherein the user desktop 1-n computers may be respectively connected to their allocated VM 1-n and operating system OS1-n. This is controlled by hypervisor 21. The ASIC compression core enables data from the user computers, desktops 1-n, to be suitably compressed so as to be most efficiently stored in association with the virtual machine platform for usage by the virtual machines to which the data is illustrated to be applied via connections 13, 14 and 15. Standard ASIC cores are described in the publication: Data compression technology in ASIC cores, S. H. Burroughs et al., in the IBM Journal of Research and Development, Volume 42, Number 6, 1998.

The tailoring of the ASIC core has sufficient flexibility so that a connection 16 may be formed from ASIC Compression Core 12 through the Baseboard Management Controller (BMC) 17 in the platform 20 directly into BIOS 22. BMC is a conventional specialized mini-controller embedded in a conventional server computer baseboard that functions as the intelligence using the conventional Intelligent Platform Management Interface (IPMI). The BMC, thus, functions to manage the interface between the system management software and the platform hardware.

As will be hereinafter described in greater detail with respect to the program of this invention, which may be stored in the hypervisor 21, the BMC 17 tracks the number of users in the environment of the virtual machine platform 20. When a session between a client device (user desktop) and its allocated VM is tracked to have been non-active for a predetermined period, the BMC removes the user computer from the activity pool. Each entry and exit from the non-activity state is logged for each client device or user computer. In an embodiment to be described with respect to Fig. 4, use is made of an embedded activity timer proved in standard ASIC cores. The BMC sets a timer when activity for a particular user computer stops. Then, upon time expiration the non-active user computer is removed from an activity tracking table located in the BMC through input to the BIOS 22. With this implementation, the tracking and communication of the active and non-active states of the user computers or client devices is carried out independent of the operating systems and hypervisor of the virtual machine platform, allowing independent control of the low power states for the virtual machine platform.

Now, with reference to Fig. 2, there will be shown the implementation for power management as applied in a system that has multiple virtual machine platforms (A-N) respectively connected to a plurality of sets of user computers (client devices). Such an arrangement is utilized in virtual machine environments used for virtual server farms. In such an arrangement, each Platform A-N may each function independently in the manner described for the virtual machine platform of Fig. 1. For convenience in illustration, elements 111 through 117 and 120 through 122 in Platform N correspond to and function in the same manner as described above for elements 11 through 17 and 20 through 22 in Platform A.

Fig. 3 is a flowchart showing the development of a process according to the present invention for power management in a virtual machine environment. In a multi-virtual machine platform environment, an implementation is provided wherein each virtual machine platform has a plurality of virtual machines, each of which is adapted to be connected to one of a plurality of user computers (client devices), step 71. Provision is made to connect the user computers of step 71 to a virtual machine via a network, step 72. Provision is made for monitoring (independently of any connection in step 71) whether each of the plurality of

computers is in the active state, step 73. Provision is made for determining if all of the plurality of connected computers are in the non-active state, step 74. Provision is made for the switching of the virtual machine platform into a reduced power consumption state responsive to a determination in step 74 that all of the connected computers are in the non-active state, step 75. Provision is made for controlling the switching of step 75 within the virtual machine platform wherein communication as to whether the connected computers are in a non-active state is by a path independent of the operating system of the virtual machine platform, step 76. Provision is made for communicating the non-active states determined in step 76 directly to the BIOS of the virtual machine platform, step 77.

The running of the process set up in Fig. 3 is described with respect to the flowchart of Fig. 4. Initially, step 81, the system has been powered on. Accordingly, there are remote user computers (client devices) connected, step 82. A counter is set (step 83):

Users Active = Remote users + 1

A determination is then made, step 84, as to whether there has been a change in remote user computer activity, step 84. If Yes, then, step 85, a determination is made as to whether there has been an interrupt (or change) in the activity of a remote user, step 85. In this step, a user can connect, disconnect or become inactive (if the user's activity timer has expired). If Yes, there has been a user activity interrupt, the remote user has disconnected from the active state, step 86. Then the counter is changed (step 87):

Users active = users active - 1

A determination is then made (step 88):

Active users = 0, step 88

If No, the flow is branched back to step 84 and the process continued. If Yes, step 89, there is a BIOS/ACIP interrupt in the virtual machine platform to allow the system into a low power "S3 Sleep" state. The BIOS then advises the hypervisor for the platform to save the

contexts, step 90. In this low power consumption sleep state, the activity of the remote user computers continues to be monitored, step 90, for a change back to a remote user activity state. If Yes, then step 91, there is a BIOS/ACIP change in the virtual machine platform into a high activity power-on "S5" full power state. The BIOS then advises the hypervisor for the platform to restore the contexts, step 92.

Although certain preferred embodiments have been shown and described, it will be understood by one of ordinary skill in the art that many changes and modifications may be made therein.

CLAIMS

1. A system for power management in a virtual machine environment comprising:

a virtual machine platform providing a plurality of virtual machines;

a plurality of separate client devices, each connected to a respective one of said virtual machines;

means, independent of said connections of said computers to said virtual machines, for determining if each of said client devices connected to said virtual machines is in an active state; and

means for switching said virtual machine platform into a reduced power consumption state when all of said client devices connected to virtual machines are in a non-active state.

2. The system for power management of claim 1, wherein:

said plurality of client devices are remote from said virtual machine platform; and

further including network means for respectively connecting said plurality of client devices to said virtual machine platform.

3. The system for power management of claim 2, wherein

said network means are World Wide Web network means.

4. The system for power management of claim 2, further including:

a plurality of said virtual machine platforms, each platform respectively providing a plurality of virtual machines; and

a plurality of sets, each set including a plurality of client devices respectively connected to the virtual machines in one of said plurality of virtual machine platforms; wherein

said means for switching switch a respective one of said virtual machine platforms into a reduced power consumption state when all of said client devices connected to virtual machines in said respective one platform are in a non-active state.

5. The system for power management of claim 4, wherein:

said plurality of said virtual machine platforms are servers arranged as a virtual server farm; and

each of said plurality of sets of client devices are client computers connected to the virtual machines in the respective servers.

6. The system for power management of claim 2, wherein:

said means for switching said virtual machine platform into said reduced power state are in said virtual machine platform; and

further including means for communicating whether each of said client devices are in a non-active state to said means for switching by a path independent of an operating system of said virtual machine platform.

7. The system for power management of claim 6, wherein said virtual machine platform includes a hypervisor and a BIOS; and

said path for communicating the non-active state is connected to the BIOS independent of said hypervisor.

8. The system for power management of claim 7, wherein: said virtual machine platform includes a baseboard management controller (BMC) having means for tracking the non-active states of said client devices; and

said path for communicating the non-active state is connected to the BIOS through said BMC.

9. A method for power management in a virtual machine platform including a plurality of virtual machines, each respectively connected to each of a plurality of separate client devices comprising:

determining, independently of said connections of said client devices to said virtual machines, if each of said client devices connected to said virtual machines is in an active state; and

switching said virtual machine platform into a reduced power consumption state when all of said client devices connected to virtual machines are in a non-active state.

10. The method of claim 9, wherein:

said plurality of client devices are remote from said virtual machine platform; and

respectively connecting said plurality of client devices to said virtual machine platform through a network.

11. The method of claim 10, wherein said network is the World Wide Web.

12. The method of claim 10 wherein said power management is for a plurality of said virtual machine platforms, each platform respectively providing a plurality of virtual machines and includes:

connecting a plurality of sets, each set including a plurality of client devices, respectively to the virtual machines in one of said plurality of virtual machine platforms;

wherein a respective one of said virtual machine platforms is switched into a reduced power consumption state when all of said client devices connected to virtual machines in said respective one platform are in a non-active state.

13. The method of claim 10, wherein

said switching of said virtual machine platform into said reduced power state is controlled in said virtual machine platform; and

further including the step of communicating whether each of said client devices are in a non-active state to said virtual machine platform by a path independent of an operating system of said virtual machine platform.

14. The method of claim 13, wherein said path for communicating the non-active state is connected to the BIOS of the virtual machine platform independent of the hypervisor of the virtual machine platform.

15. A computer program comprising computer program code to, when loaded into a computer system and executed thereon, cause said computer to perform all the steps of the method of any of claims 9 to 14.

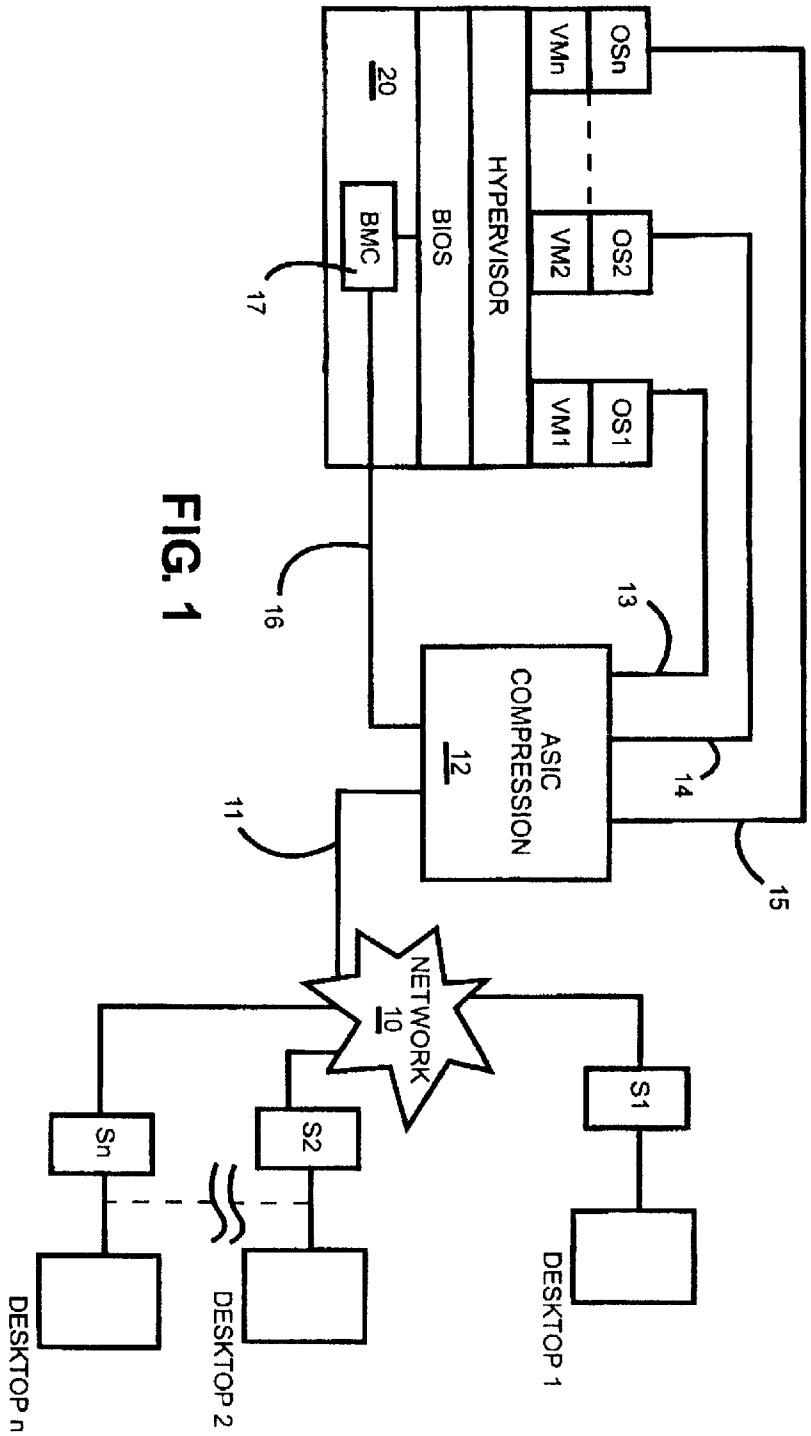
VMn
OSn
20
HYPERVISOR
BIOS
BMC
VM2
OS2
VM1
OS1
17
16
13
14
15
11
ASIC
COMPRESSION
12
NETWORK
10
Sn
S2
S1
DESKTOP n
DESKTOP 2
DESKTOP 1

FIG. 1

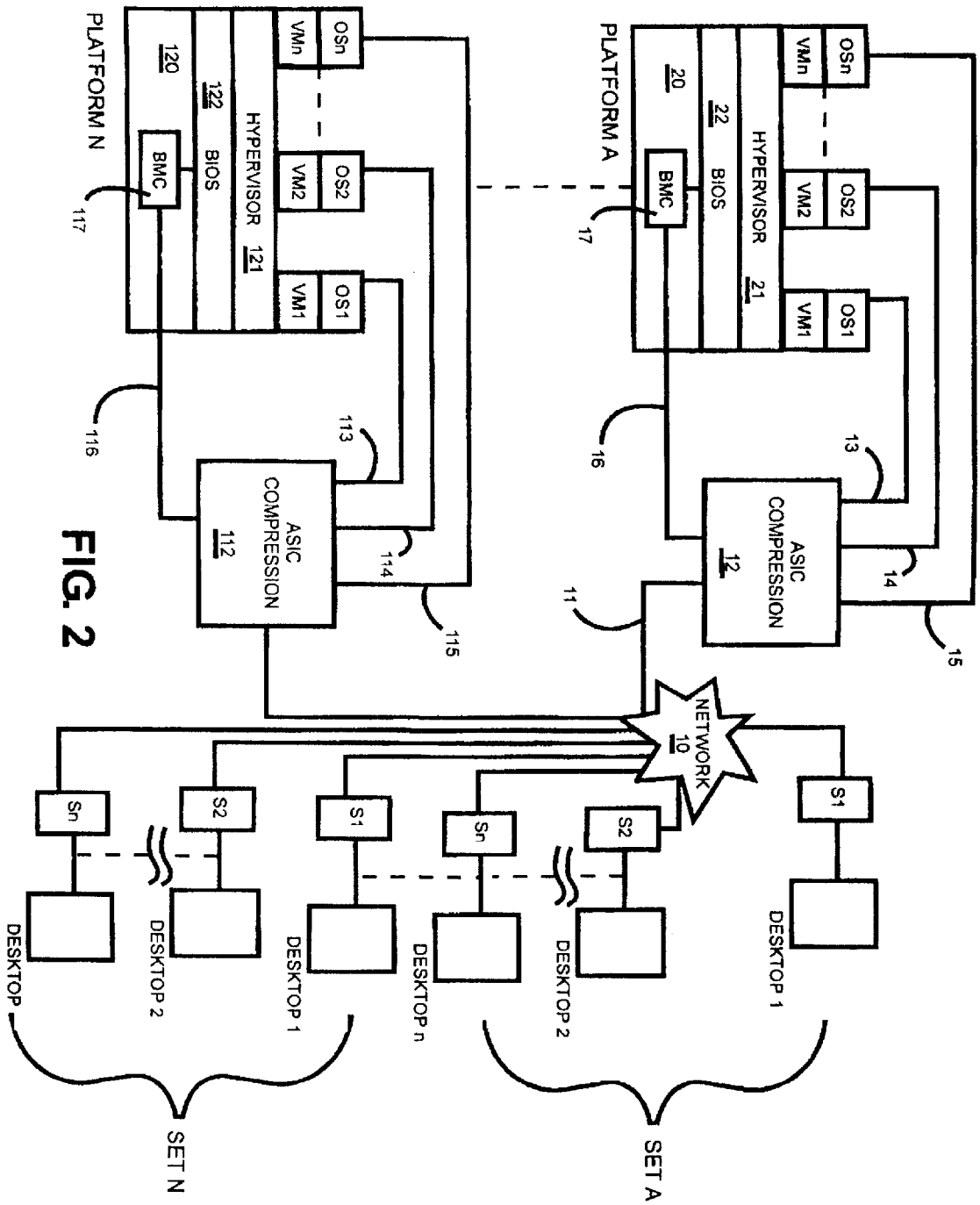
PLATFORM A
PLATFORM N
HYPERVISOR
BIOS
BMC
ASIC COMPRESSION
NETWORK
VMn
VM2
VM1
OSn
OS2
OS1
SET A
SET N
DESKTOP
DESKTOP 1
DESKTOP 2
DESKTOP n

FIG. 2

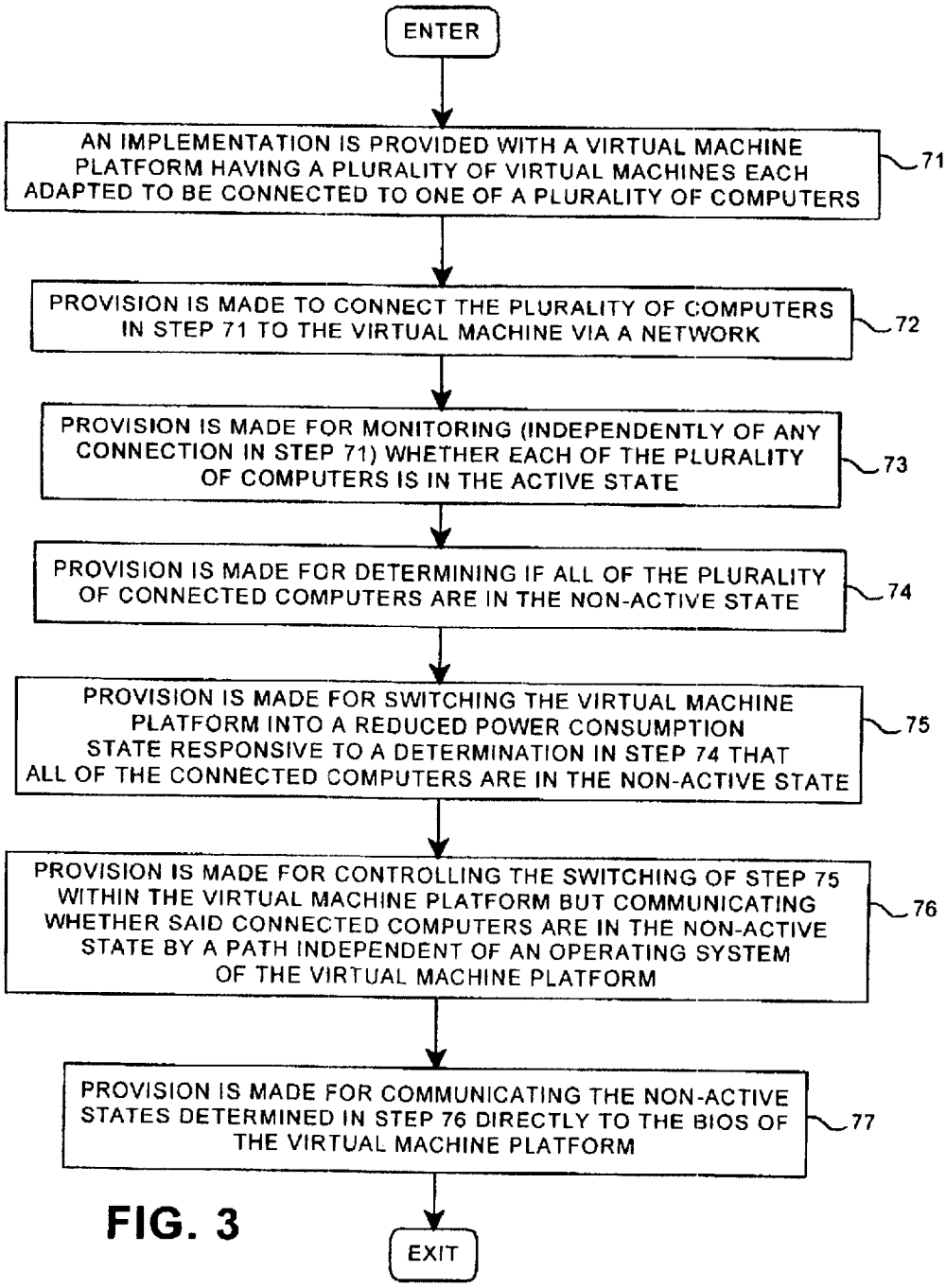
ENTER
AN IMPLEMENTATION IS PROVIDED WITH A VIRTUAL MACHINE PLATFORM HAVING A PLURALITY OF VIRTUAL MACHINES EACH ADAPTED TO BE CONNECTED TO ONE OF A PLURALITY OF COMPUTERS
71
PROVISION IS MADE TO CONNECT THE PLURALITY OF COMPUTERS IN STEP 71 TO THE VIRTUAL MACHINE VIA A NETWORK
72
PROVISION IS MADE FOR MONITORING (INDEPENDENTLY OF ANY CONNECTION IN STEP 71) WHETHER EACH OF THE PLURALITY OF COMPUTERS IS IN THE ACTIVE STATE
73
PROVISION IS MADE FOR DETERMINING IF ALL OF THE PLURALITY OF CONNECTED COMPUTERS ARE IN THE NON-ACTIVE STATE
74
PROVISION IS MADE FOR SWITCHING THE VIRTUAL MACHINE PLATFORM INTO A REDUCED POWER CONSUMPTION STATE RESPONSIVE TO A DETERMINATION IN STEP 74 THAT ALL OF THE CONNECTED COMPUTERS ARE IN THE NON-ACTIVE STATE
75
PROVISION IS MADE FOR CONTROLLING THE SWITCHING OF STEP 75 WITHIN THE VIRTUAL MACHINE PLATFORM BUT COMMUNICATING WHETHER SAID CONNECTED COMPUTERS ARE IN THE NON-ACTIVE STATE BY A PATH INDEPENDENT OF AN OPERATING SYSTEM OF THE VIRTUAL MACHINE PLATFORM
76
PROVISION IS MADE FOR COMMUNICATING THE NON-ACTIVE STATES DETERMINED IN STEP 76 DIRECTLY TO THE BIOS OF THE VIRTUAL MACHINE PLATFORM
77
EXIT

FIG. 3

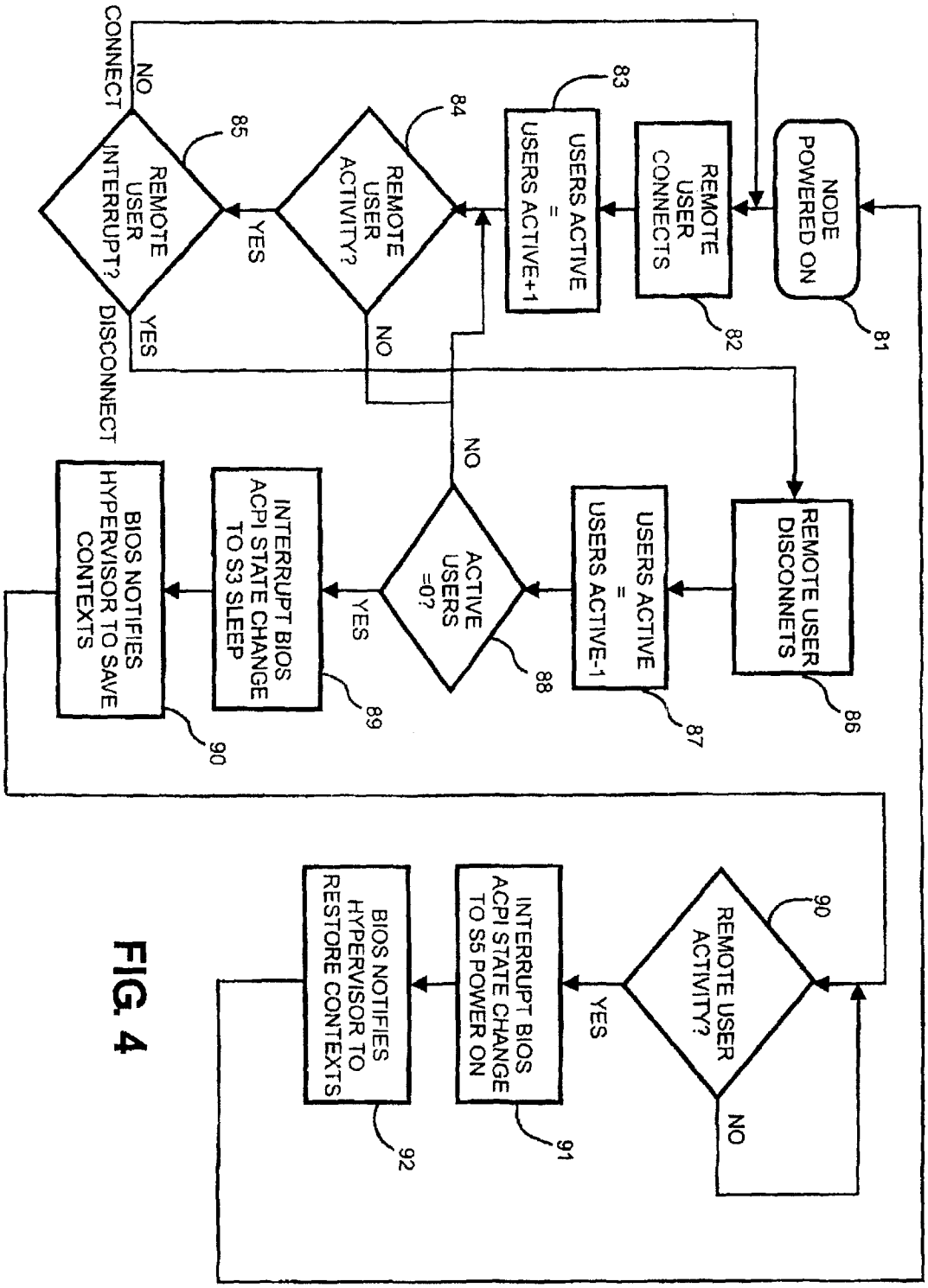
NODE POWERED ON
81
REMOTE USER CONNECTS
82
USERS ACTIVE = USERS ACTIVE+1
83
REMOTE USER ACTIVITY?
84
NO
YES
REMOTE USER INTERRUPT?
85
YES
NO
DISCONNECT
CONNECT
REMOTE USER DISCONNETS
86
USERS ACTIVE = USERS ACTIVE-1
87
ACTIVE USERS =0?
88
NO
YES
INTERRUPT BIOS ACPI STATE CHANGE TO S3 SLEEP
89
BIOS NOTIFIES HYPERVISOR TO SAVE CONTEXTS
90
REMOTE USER ACTIVITY?
90
NO
YES
INTERRUPT BIOS ACPI STATE CHANGE TO S5 POWER ON
91
BIOS NOTIFIES HYPERVISOR TO RESTORE CONTEXTS
92

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/051856

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F1/32
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2002/083110 A1 (KOZUCH MICHAEL [US] ET AL) 27 June 2002 (2002-06-27) paragraph [0016] - paragraph [0034]; figures 1-4	1-15
X,P	US 2009/217072 A1 (GEBHART ALEXANDER [DE] ET AL) 27 August 2009 (2009-08-27) abstract paragraph [0016] - paragraph [0045]; figures 1-3	1-15
A	JP 2006 172283 A (CANON KK) 29 June 2006 (2006-06-29) abstract	1,6-9, 13-15
A	US 2008/177424 A1 (WHEELER ANDREW R [US]) 24 July 2008 (2008-07-24) paragraph [0008] - paragraph [0031]	1-5, 9-12,15

-/--

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

[X] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 May 2010	01/06/2010

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/051856

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2004 086344 A (CANON KK) 18 March 2004 (2004-03-18) abstract -----	1,9,15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2010/051856

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2002083110 A1	27-06-2002	NONE	
US 2009217072 A1	27-08-2009	NONE	
JP 2006172283 A	29-06-2006	NONE	
US 2008177424 A1	24-07-2008	CN 101231552 A	30-07-2008
JP 2004086344 A	18-03-2004	NONE	