



(51) International Patent Classification:
G06F 9/06 (2006.01) **G06F 9/44** (2006.01)

(21) International Application Number:
PCT/US2009/036297

(22) International Filing Date:
6 March 2009 (06.03.2009)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US):
HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 20555 S. H. 249, Houston, TX 77070 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **PROVENCHER, Michael, A.** [US/US]; Hp, Houston, Tx, (cca08) 6 8670, Mail Stop 080610, 11445 Compaq Center Dr. W., Houston, TX 77070 (US).

(74) Agents: **CZARNECKI, Michael, S.** et al.; Hewlett-packard Company, Intellectual Property Administration, Mail Stop 35, P.o. Box 272400, Fort Collins, CO 80527-2400 (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, BR, BW, BY, BZ, CA, CH, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, 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, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

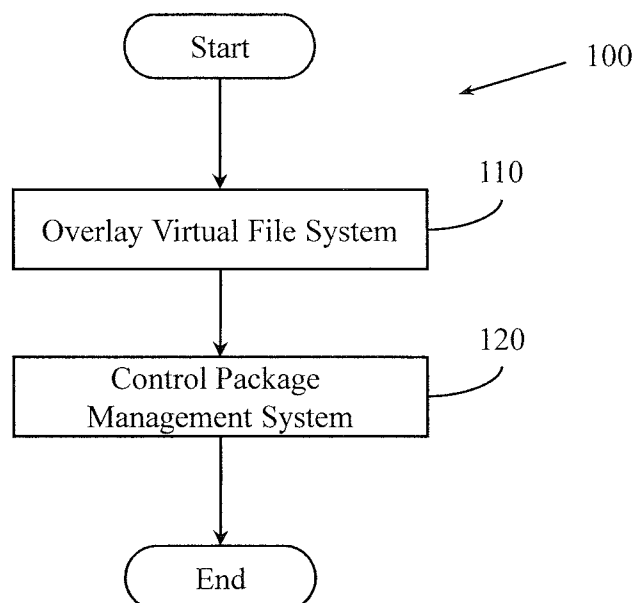
Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

[Continued on next page]

(54) Title: APPLICATION SEQUENCING

Figure 1



(57) Abstract: Systems, methods, and other embodiments associated with application sequencing are described. One example method includes overlaying (110) a virtual file system on an underlying directory structure on a computer. The example method may also include controlling (120) a package management system on the computer to sequence an application. Data associated with the application may be stored on the virtual file system.

WO 2010/101574 A1



Published:

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

APPLICATION SEQUENCING

BACKGROUND

5 **[0001]** Companies sometimes employ workers who have specialized tasks. These workers may use subsets of software available to the company based on their tasks. For example, a programmer may use a word processor, a spreadsheet application, a text editor, a compiler, a secure test environment, and so forth. An artist may use an image editor, a 3D modeler, a word processor, a graphics tablet
10 interface, a publishing application, and so forth. A financial analyst may use a word processor, an analysis tool, a web browser, a spreadsheet application, and so forth. Previously, applications that an employee used were installed on the employee's computer. However, when updates to an application were made, and/or a new version came out, individual computers on which the application was
15 installed may have needed to be updated. This may have required a technician to sit down at and/or remotely control computers individually to perform an update. To reduce labor costs and downtime losses, companies have begun moving towards application streaming. Application streaming involves hosting applications on a centralized server and providing applications to users over a network as they
20 are requested by the users.

[0002] Some conventional implementations of application streaming, sometimes referred to as application sequencing, use virtualization techniques to provide an

application to a user. These implementations may allow a user to begin to use an application after only a portion of the application's functionality has been downloaded. For example, some sophisticated spreadsheet applications allow users to generate graphs from data that has been input. However, this feature may not be necessary to view and/or input data into a basic spreadsheet. Thus application streaming may allow a user to begin using an application before the user's computer has received the entire application. However, conventional application sequencing techniques that rely on virtualization typically burden streamed applications with runtime hooks that may have to be configured into an application launcher. Runtime hooks may allow the application to ensure that resources that the application uses are available. Additionally, conventional virtualization techniques may not fully isolate streamed applications from a system.

15 BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate various example systems, methods, and other example embodiments of various aspects of the invention. It will be appreciated that the illustrated element boundaries (e.g., boxes, groups of boxes, or other shapes) in the figures represent one example of the boundaries. One of ordinary skill in the art will appreciate that in some examples one element may be designed as multiple elements or that multiple elements may be designed as one element. In some examples, an element shown as an internal component of another element may be implemented as an external component and vice versa. Furthermore, elements may not be drawn to scale.

[0004] Figure 1 illustrates an example embodiment of a method associated with application sequencing.

[0005] Figure 2 illustrates another example embodiment of a method associated with application sequencing.

[0006] Figure 3 illustrates an example embodiment of a system associated with application streaming.

5 **[0007]** Figure 4 illustrates another example embodiment of a system associated with application streaming.

[0008] Figure 5 illustrates an example computing environment in which example systems and methods, and equivalents, may operate.

10 SUMMARY

[0009] Systems and methods associated with application streaming are described. One example embodiment includes a computer-readable medium that stores computer-executable instructions that when executed by a computer cause the computer to perform a method. The method comprises overlaying a virtual file system on an underlying directory structure on a computer. The method also includes controlling a package management system on the computer to sequence an application. In this example, data associated with the application is stored on the virtual file system.

20 DETAILED DESCRIPTION

[0010] Systems and methods associated with application streaming are described. One example method includes overlaying a virtual file system on top of an underlying directory structure (e.g., a file system on a hard disk). This may facilitate preventing changes that are made to the virtual file system from directly affecting the underlying directory structure. The example method also includes controlling a package management system to sequence an application. The

package management system may be controlled to store data associated with the application to the virtual file system.

[0011] By virtualizing the file system on which application data is sequenced, the application may be effectively isolated from the base system. This may allow the system to be reset to a known state without having to worry about what has been done in or by the application. Further, the virtual file system may provide read access to data stored in the underlying directory structure. This may give the application access to data and/or resources stored in the underlying directory structure. By sequencing the application to the virtual file system with a package management system, the package management system may automatically determine what additional resources the application may require and automatically ensure that these resources are available to the application. This may include downloading and/or installing additional data and/or applications. Thus, the package management system may reduce the number of runtime hooks that are built into a sequenced application that an administrator may have otherwise had to pre-configure. The combination of a virtual file system overlay and a package management system facilitates application streaming in a manner not previously contemplated as package management systems are typically associated with configuring and maintaining data and applications that are installed on non-virtualized data structures.

[0012] By way of illustration, a package management system may compare a list of installed packages to a set of rules for dependencies, recommendations and/or suggestions specified in a requested application's packaging. Dependencies may specify other packages that the application may require to function properly. Recommendations may include secondary features of the application. Suggestions may include convenience features (e.g., macros, a more sophisticated user interface, and so forth). Thus, the package management system may automatically sequence the requested application and its dependencies without an administrator configuring the sequenced application in

advance. The package management system may also be able to ensure that specific hardware requirements of an application are available to the application based on a preconfigured policy.

[0013] The following includes definitions of selected terms employed herein.

5 The definitions include various examples and/or forms of components that fall within the scope of a term and that may be used for implementation. The examples are not intended to be limiting. Both singular and plural forms of terms may be within the definitions.

[0014] References to “one embodiment”, “an embodiment”, “one example”, “an example”, and so on, indicate that the embodiment(s) or example(s) so described may include a particular feature, structure, characteristic, property, element, or limitation, but that not every embodiment or example necessarily includes that particular feature, structure, characteristic, property, element or limitation. Furthermore, repeated use of the phrase “in one embodiment” does not necessarily
15 refer to the same embodiment, though it may.

[0015] ASIC: application specific integrated circuit.

[0016] CD: compact disk.

[0017] CD-R: CD recordable.

[0018] CD-RW: CD rewriteable.

20 **[0019]** DVD: digital versatile disk and/or digital video disk.

[0020] HTTP: hypertext transfer protocol.

[0021] PCI: peripheral component interconnect.

[0022] PCIe: PCI express.

[0023] RAM: random access memory.

25 **[0024]** DRAM: dynamic RAM.

[0025] SRAM: static RAM.

[0026] ROM: read only memory.

[0027] PROM: programmable ROM.

[0028] USB: universal serial bus.

[0029] LAN: local area network.

5 [0030] WAN: wide area network.

[0031] "Computer component", as used herein, refers to a computer-related entity (e.g., hardware, firmware, software in execution, combinations thereof). Computer components may include, for example, a process running on a processor, a processor, an object, an executable, a thread of execution, and a
10 computer. A computer component(s) may reside within a process and/or thread. A computer component may be localized on one computer and/or may be distributed between multiple computers.

[0032] "Computer communication", as used herein, refers to a communication between computing devices (e.g., computer, personal digital assistant, cellular
15 telephone) and can be, for example, a network transfer, a file transfer, an applet transfer, an email, an HTTP transfer, and so on. A computer communication can occur across, for example, a wireless system (e.g., IEEE 802.11), an Ethernet system (e.g., IEEE 802.3), a token ring system (e.g., IEEE 802.5), a LAN, a WAN, a point-to-point system, a circuit switching system, a packet switching system, and
20 so on.

[0033] "Computer-readable medium", as used herein, refers to a medium that stores signals, instructions and/or data. A computer-readable medium may take forms, including, but not limited to, non-volatile media, and volatile media. Non-volatile media may include, for example, optical disks, magnetic disks, and so on.
25 Volatile media may include, for example, semiconductor memories, dynamic memory, and so on. Common forms of a computer-readable medium may include, but are not limited to, a floppy disk, a flexible disk, a hard disk, a magnetic tape, other magnetic medium, an ASIC, a CD, other optical medium, a RAM, a ROM, a

memory chip or card, a memory stick, and other media from which a computer, a processor or other electronic device can read.

[0034] “Logic”, as used herein, includes but is not limited to hardware, firmware, software in execution on a machine, and/or combinations of each to perform a function(s) or an action(s), and/or to cause a function or action from another logic, method, and/or system. Logic may include a software controlled microprocessor, a discrete logic (e.g., ASIC), an analog circuit, a digital circuit, a programmed logic device, a memory device containing instructions, and so on. Logic may include one or more gates, combinations of gates, or other circuit components. Where multiple logical logics are described, it may be possible to incorporate the multiple logical logics into one physical logic. Similarly, where a single logical logic is described, it may be possible to distribute that single logical logic between multiple physical logics.

[0035] An “operable connection”, or a connection by which entities are “operably connected”, is one in which signals, physical communications, and/or logical communications may be sent and/or received. An operable connection may include a physical interface, an electrical interface, and/or a data interface. An operable connection may include differing combinations of interfaces and/or connections sufficient to allow operable control. For example, two entities can be operably connected to communicate signals to each other directly or through one or more intermediate entities (e.g., processor, operating system, logic, software). Logical and/or physical communication channels can be used to create an operable connection.

[0036] “Signal”, as used herein, includes but is not limited to, electrical signals, optical signals, analog signals, digital signals, data, computer instructions, processor instructions, messages, a bit, a bit stream, or other means that can be received, transmitted and/or detected.

[0037] “Software”, as used herein, includes but is not limited to, one or more executable instruction that cause a computer, processor, or other electronic device

to perform functions, actions and/or behave in a desired manner. "Software" does not refer to stored instructions being claimed as stored instructions per se (e.g., a program listing). The instructions may be embodied in various forms including routines, algorithms, modules, methods, threads, and/or programs including
5 separate applications or code from dynamically linked libraries.

[0038] "User", as used herein, includes but is not limited to one or more persons, software, computers or other devices, or combinations of these.

[0039] Some portions of the detailed descriptions that follow are presented in terms of algorithms and symbolic representations of operations on data bits within
10 a memory. These algorithmic descriptions and representations are used by those skilled in the art to convey the substance of their work to others. An algorithm, here and generally, is conceived to be a sequence of operations that produce a result. The operations may include physical manipulations of physical quantities. Usually, though not necessarily, the physical quantities take the form of electrical or
15 magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a logic, and so on. The physical manipulations create a concrete, tangible, useful, real-world result.

[0040] It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters,
20 terms, numbers, and so on. It should be borne in mind, however, that these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise, it is appreciated that throughout the description, terms including processing, computing, determining, and so on, refer to actions and processes of a
25 computer system, logic, processor, or similar electronic device that manipulates and transforms data represented as physical (electronic) quantities.

[0041] Example methods may be better appreciated with reference to flow diagrams. While for purposes of simplicity of explanation, the illustrated methodologies are shown and described as a series of blocks, it is to be

appreciated that the methodologies are not limited by the order of the blocks, as some blocks can occur in different orders and/or concurrently with other blocks from that shown and described. Moreover, less than all the illustrated blocks may be required to implement an example methodology. Blocks may be combined or
5 separated into multiple components. Furthermore, additional and/or alternative methodologies can employ additional, not illustrated blocks.

[0042] Figure 1 illustrates an example embodiment of a method 100 associated with application sequencing. Method 100 includes, at 110, overlaying a virtual file system on an underlying directory structure. In one example, the virtual file system
10 may be configured to prevent data written to the virtual file system from being written to the underlying directory structure. This may prevent changes made to resources used by the virtual file system from affecting a state of the underlying directory structure. The underlying directory structure may be built into a file system residing on a computer that is performing method 100. Method 100 also
15 includes, at 120, controlling a package management system on the computer to sequence an application. Data associated with the application may be stored on the virtual file system.

[0043] Sequencing the application may comprise providing application request data to a remote server. The application request data may identify the application
20 being sequenced. In one example, the application request data may identify the application with data associated with a requesting entity. For example, a requesting entity identified as an artist may cause a 3D modeling application to be sequenced while a requesting entity identified as a programmer may cause a scripting application to be sequenced. Sequencing the application may also
25 comprise receiving a first portion of the application from the remote server. In one example, the first portion of the application may comprise a minimum portion of the application. This minimum portion of the application may allow a primary function of the application to operate. For example, a minimum portion of a word processor

application may comprise instructions that cause a computer to display a simple user interface and cause the computer to facilitate manipulation of a text file.

[0044] Sequencing the application may also comprise receiving a second portion of the application from the remote server. In one example, the second portion may be downloaded based on information in the application request data. By way of illustration, the application request data may identify a role or title of a user requesting the sequencing of the application. A second portion of a spreadsheet application downloaded for a financial analyst may include a graphical analysis feature that allows the financial analyst to create and manipulate graphs. A second portion of a spreadsheet application downloaded for a publicist may include a mail merge feature that allows the publicist to merge address data with a boilerplate letter to create personalized mailings. In another example, the second portion may be downloaded in response to receiving a request for access to a feature that is a secondary function of the application. Thus, the second portion may comprise a portion of the application that contains instructions that cause the computer to facilitate providing access to the secondary feature. By way of illustration, a professor using a slideshow application may receive a portion of the slideshow application that allows the professor to play a slideshow that he has created. If the professor determines he needs to modify a slide, a portion of the slideshow application that facilitates editing slides may be provided on demand.

[0045] While Figure 100 illustrates various actions occurring in serial, it is to be appreciated that various actions illustrated in Figure 100 could occur substantially in parallel. By way of illustration, a first process could overlay the virtual file system, and a second process could control the package management system. While two processes are described, it is to be appreciated that a greater and/or lesser number of processes could be employed and that lightweight processes, regular processes, threads, and other approaches could be employed.

[0046] In one example, a method may be implemented as computer executable instructions. Thus, in one example, a computer-readable medium may store

computer executable instructions that if executed by a machine (e.g., processor) cause the machine to perform a method. While executable instructions associated with the above method are described as being stored on a computer-readable medium, it is to be appreciated that executable instructions associated with other
5 example methods described herein may also be stored on a computer-readable medium.

[0047] Figure 2 illustrates an example embodiment of a method 200 associated with application sequencing. Method 200 includes several actions similar to those described in connection with method 100 (figure 1). For example, method 200
10 includes overlaying a virtual file system at 210, and controlling a package management system at 220. However, method 200 includes additional actions.

[0048] Method 200 includes, at 230, copying data from the virtual file system to a non-virtual memory. The non-virtual memory may be the underlying directory structure, a local storage medium, a remote storage medium, and so on. Copying
15 data to a non-virtual memory may facilitate backing up data that was manipulated while the application was running. For example, a file a user has created or modified may be lost if data associated with the file is not copied to a non-virtual memory. Method 200 also includes, at 240, terminating the virtual file system. Terminating the virtual file system may restore a computer on which the virtual file
20 system was created to a prior state. The prior state may be a known state that facilitates operations of future instances of application sequencing methods. The virtual file system may be terminated as a result of rebooting a system with which the underlying directory structure is associated, terminating an instance of the application, a signal from an administrating entity, a time limit expiring, and so forth.
25 Thus method 200 illustrates how an application may be sequenced to a virtual file system, and how data from the virtual file system may be treated when and/or before the application is no longer being used.

[0049] Figure 3 illustrates an example embodiment of a system 300 associated with application sequencing. System 300 includes a first file system 310. First file

system 310 may comprise a directory structure. System 300 also includes an overlay logic 320. Overlay logic 320 may overlay a virtual file system on top of the directory structure. In one example, the virtual file system may prevent data written to the virtual file system from being written to first file system 310. System 300 also includes a package management logic 330. Package management logic 330 may sequence an application. Data associated with the application may be stored on the virtual file system. Thus figure 3 illustrates a system 300 that can virtualize a file system instead of virtualizing an application. This facilitates reliably restoring the base system when the application is no longer being used on the system.

[0050] Figure 4 illustrates an example embodiment of a system 400 associated with application sequencing. System 400 includes several elements similar to those described in connection with system 300 (figure 3). For example, system 400 includes a first file system 410, an overlay logic 420, and a package management logic 430. System 400 also includes a control logic 440. Control logic 440 may control overlay logic 420 to overlay the virtual file system and package management logic 430 to sequence the application. Control logic 440 may also initiate the application. In one example control logic 440 may activate in response to a user input. By way of illustration a desktop icon may allow a user to cause control logic 440 to initiate operation so that the user may use an application with which the desktop icon is associated. Thus system 400 illustrates how a user input may be used to initiate application sequencing that uses a virtual file system.

[0051] Figure 5 illustrates an example computing device in which example systems and methods described herein, and equivalents, may operate. The example computing device may be a computer 500 that includes a processor 502, a memory 504, and input/output ports 510 operably connected by a bus 508. In one example, the computer 500 may include an application streaming logic 530 configured to facilitate sequencing applications. In different examples, the logic 530 may be implemented in hardware, software, firmware, and/or combinations thereof. While the logic 530 is illustrated as a hardware component attached to the

bus 508, it is to be appreciated that in one example, the logic 530 could be implemented in the processor 502.

[0052] Thus, logic 530 may provide means (e.g., hardware, software, firmware) for overlaying a virtual file system on top of a directory structure. The means may be implemented, for example, as an ASIC. The means may also be implemented as computer executable instructions that are presented to computer 500 as data 516 that are temporarily stored in memory 504 and then executed by processor 502. Logic 530 may also provide means (e.g., hardware, software, firmware) for sequencing an application to the virtual file system. Data associated with the application may be stored on the virtual file system.

[0053] Generally describing an example configuration of the computer 500, the processor 502 may be a variety of various processors including dual microprocessor and other multi-processor architectures. A memory 504 may include volatile memory and/or non-volatile memory. Non-volatile memory may include, for example, ROM, PROM, and so on. Volatile memory may include, for example, RAM, SRAM, DRAM, and so on.

[0054] A disk 506 may be operably connected to the computer 500 via, for example, an input/output interface (e.g., card, device) 518 and an input/output port 510. The disk 506 may be, for example, a magnetic disk drive, a solid state disk drive, a floppy disk drive, a tape drive, a Zip drive, a flash memory card, a memory stick, and so on. Furthermore, the disk 506 may be a CD-ROM drive, a CD-R drive, a CD-RW drive, a DVD ROM, and so on. The memory 504 can store a process 514 and/or a data 516, for example. The disk 506 and/or the memory 504 can store an operating system that controls and allocates resources of the computer 500.

[0055] The bus 508 may be a single internal bus interconnect architecture and/or other bus or mesh architectures. While a single bus is illustrated, it is to be appreciated that the computer 500 may communicate with various devices, logics, and peripherals using other busses (e.g., PCIE, 1394, USB, Ethernet). The bus

508 can be types including, for example, a memory bus, a memory controller, a peripheral bus, an external bus, a crossbar switch, and/or a local bus.

[0056] The computer 500 may interact with input/output devices via the I/O interfaces 518 and the input/output ports 510. Input/output devices may be, for example, a keyboard, a microphone, a pointing and selection device, cameras, video cards, displays, the disk 506, the network devices 520, and so on. The input/output ports 510 may include, for example, serial ports, parallel ports, and USB ports.

[0057] The computer 500 can operate in a network environment and thus may be connected to the network devices 520 via the I/O interfaces 518, and/or the I/O ports 510. Through the network devices 520, the computer 500 may interact with a network. Through the network, the computer 500 may be logically connected to remote computers. Networks with which the computer 500 may interact include, but are not limited to, a LAN, a WAN, and other networks.

[0058] While example systems, methods, and so on have been illustrated by describing examples, and while the examples have been described in considerable detail, it is not the intention of the applicants to restrict or in any way limit the scope of the appended claims to such detail. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the systems, methods, and so on described herein. Therefore, the invention is not limited to the specific details, the representative apparatus, and illustrative examples shown and described. Thus, this application is intended to embrace alterations, modifications, and variations that fall within the scope of the appended claims.

[0059] To the extent that the term “includes” or “including” is employed in the detailed description or the claims, it is intended to be inclusive in a manner similar to the term “comprising” as that term is interpreted when employed as a transitional word in a claim.

[0060] To the extent that the term “or” is employed in the detailed description or claims (e.g., A or B) it is intended to mean “A or B or both”. When the applicants intend to indicate “only A or B but not both” then the term “only A or B but not both” will be employed. Thus, use of the term “or” herein is the inclusive, and not the
5 exclusive use. See, Bryan A. Garner, A Dictionary of Modern Legal Usage 624 (2d. Ed. 1995).

[0061] To the extent that the phrase “one or more of, A, B, and C” is employed herein, (e.g., a data store configured to store one or more of, A, B, and C) it is intended to convey the set of possibilities A, B, C, AB, AC, BC, and/or ABC (e.g.,
10 the data store may store only A, only B, only C, A&B, A&C, B&C, and/or A&B&C). It is not intended to require one of A, one of B, and one of C. When the applicants intend to indicate “at least one of A, at least one of B, and at least one of C”, then the phrasing “at least one of A, at least one of B, and at least one of C” will be employed.

Claims

What is claimed is:

1. A computer-readable medium storing computer-executable instructions that when executed by a computer cause the computer to perform a method, the method comprising:
 - overlaying a virtual file system on an underlying directory structure on the computer; and
 - controlling a package management system on the computer to sequence an application, where data associated with the application is stored on the virtual file system.
2. The computer-readable medium of claim 1, where the virtual file system prevents data written to the virtual file system from being written to the underlying directory structure.
3. The computer-readable medium of claim 1, where sequencing the application comprises:
 - providing application request data to a remote server, where the application request data identifies the application; and
 - receiving a first portion of the application from the remote server.
4. The computer-readable medium of claim 3, where the first portion of the application comprises a minimum portion of the application that allows a primary function of the application to operate.
5. The computer-readable medium of claim 4, where sequencing the application comprises:

receiving a second portion of the application from the remote server.

6. The computer-readable medium of claim 5, where the second portion is downloaded based on information in the application request data.
7. The computer-readable medium of claim 5, where the second portion is downloaded in response to receiving a request for access to a feature that is a secondary function of the application, and where the second portion comprises a portion of the application that provides access to the feature.
8. The computer-readable medium of claim 3, where the application request data identifies the application with data associated with a requesting entity.
9. The computer-readable medium of claim 1, the method comprising:
terminating the virtual file system.
10. The computer-readable medium of claim 9, where the virtual file system is terminated as a result of one of, rebooting a system with which the underlying directory structure is associated, terminating an instance of the application, a signal from an administrating entity, and a time limit expiring.
11. The computer-readable medium of claim 1, the method comprising:
copying data on the virtual file system to a non-virtual memory, where the non-virtual memory is one of, the underlying directory structure, a local storage medium, and a remote storage medium.
12. A system, comprising:
a first file system comprising a directory structure;

an overlay logic to overlay a virtual file system on the directory structure, where the virtual file system prevents data written to the virtual file system from being written to the first file system; and

a package management logic to sequence an application, where data associated with the application is stored on the virtual file system.

13. The system of claim 12, comprising:

a control logic to control the overlay logic to overlay the virtual file system, to control the package management logic to sequence the application, and to initiate the application.

14. The system of claim 13, where the control logic activates in response to a user input.

15. A system, comprising:

means for overlaying a virtual file system on a directory structure; and

means for sequencing an application to the virtual file system, where data associated with the application is stored on the virtual file system.

1/5

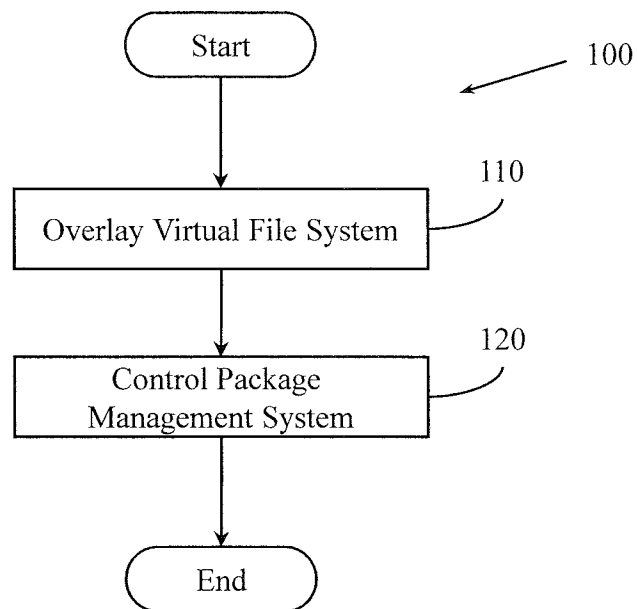


Figure 1

2/5

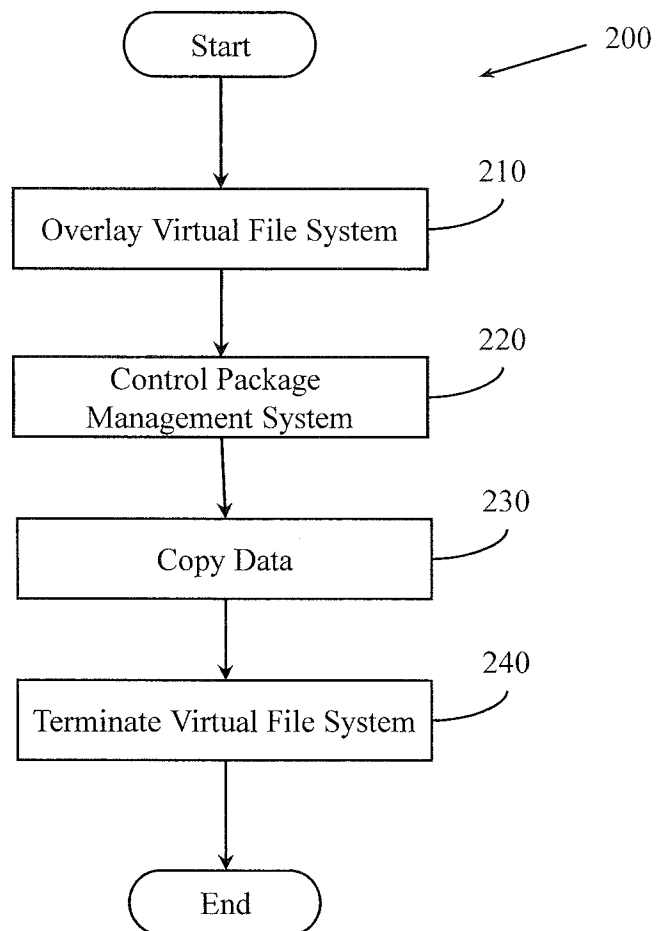


Figure 2

3/5

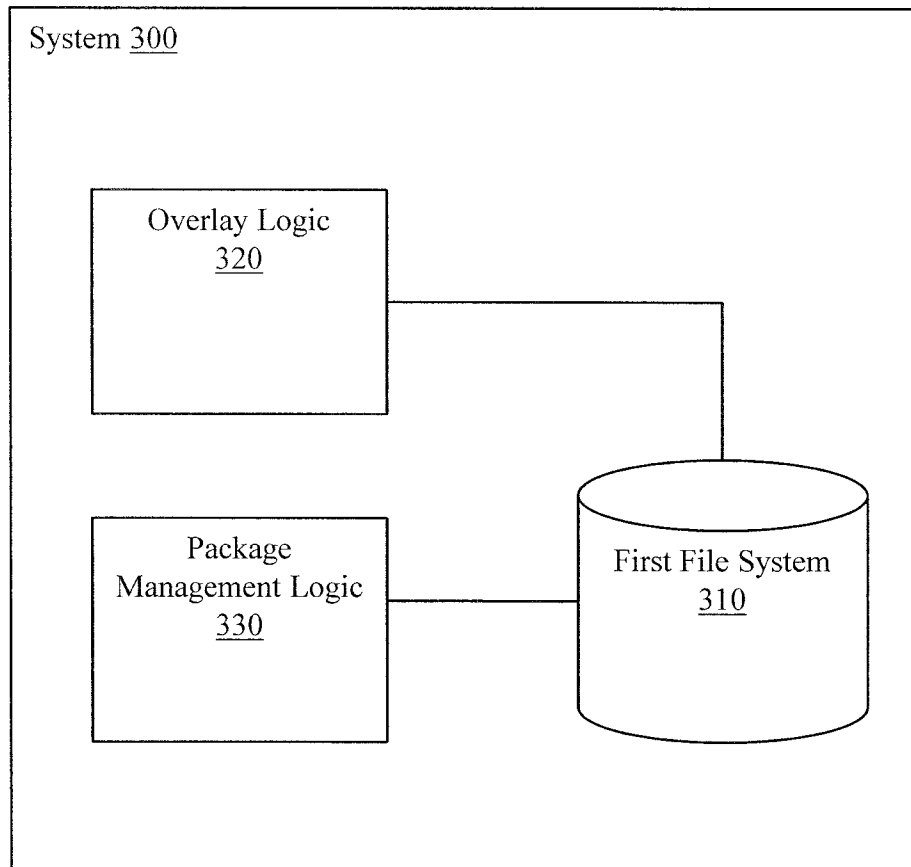


Figure 3

4/5

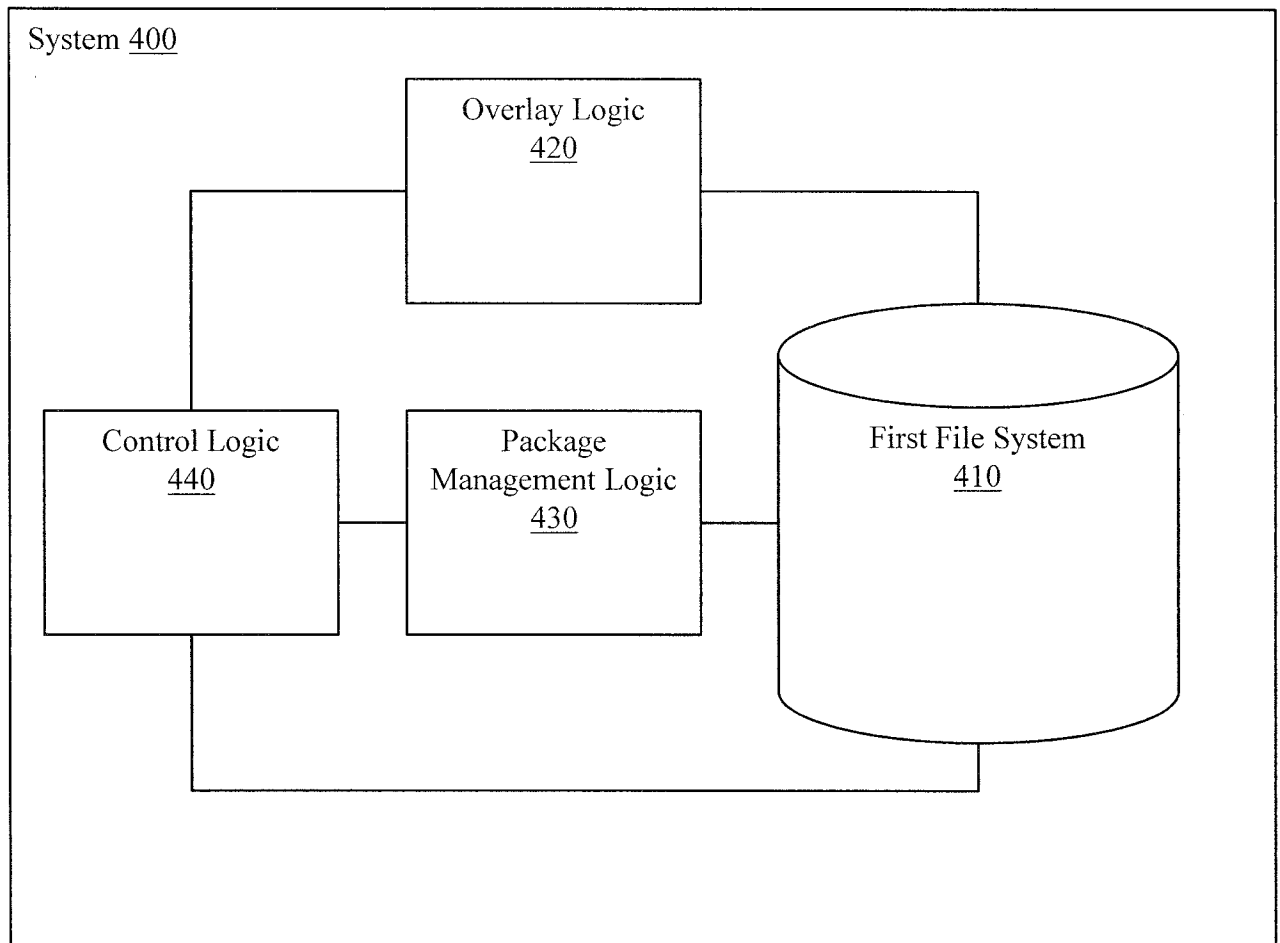


Figure 4

5/5

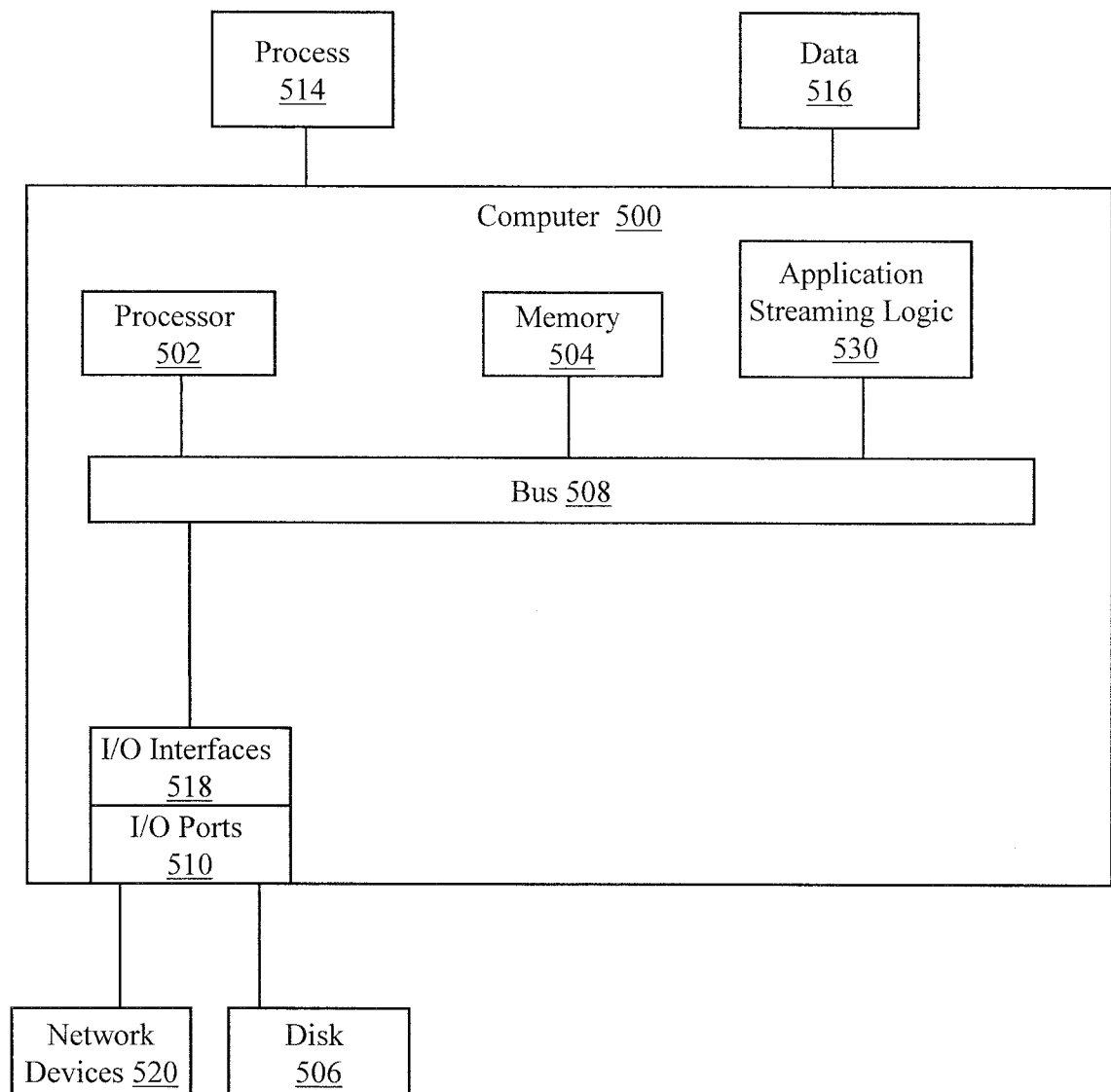


Figure 5

A. CLASSIFICATION OF SUBJECT MATTER***G06F 9/06(2006.01)i, G06F 9/44(2006.01)i***

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)

IPC8 : G06F

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

Korean Utility models and applications for Utility models since 1975

Japanese Utility models and applications for Utility models since 1975

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

eKOMPASS(KIPO internal) "virtual", "file", "overlying"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2001/0034736 A1 (DAN EYLON, et al.) 25 October 2001 See [0015] - [0074].	1-15
A	US 6393437 B1 (ERIC K. ZINDA, et al.) 21 May 2002 See column 2, line 37 - column 18, line 6.	1-15
A	US 2005/0207732 A1 (YONG-BUM CHO, et al.) 22 September 2005 See [0022] - [0061].	1-15

☐ 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

"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 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

27 NOVEMBER 2009 (27.11.2009)

Date of mailing of the international search report

27 NOVEMBER 2009 (27.11.2009)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-
gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

KWON, Oh Seong

Telephone No. 82-42-481-8526



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2009/036297

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2001-0034736 A1	25.10.2001	AU 2001-93088 A1	25.09.2001
		AU 2001-93087 A1	25.09.2001
		AU 2001-32825 A1	17.01.2001
		AU 2001-29575 A1	17.01.2001
		AU 1999-51046 A1	15.07.1999
		CA 2361342 A1	03.02.2000
		EP 1097416 A4	02.05.2007
		EP 1330738 A4	17.05.2006
		EP 1332432 A4	17.05.2006
		EP 1097416 A2	09.05.2001
		EP 1330738 A1	30.07.2003
		EP 1332432 A1	06.08.2003
		JP 2002-521745 A	16.07.2002
		US 2003-0140160 A1	24.07.2003
		US 2002-0138640 A1	26.09.2002
		US 2002-0042833 A1	11.04.2002
		US 2001-0044850 A1	22.11.2001
		US 2001-0037400 A1	01.11.2001
		US 2001-0037399 A1	01.11.2001
		US 06311221 B1	30.10.2001
		US 06574618 B2	03.06.2003
		US 07197570 B2	27.03.2007
		US 07606924 B2	20.10.2009
		WO 2002-027492 A1	04.04.2002
		WO 2001-054375 A3	26.07.2001
		WO 2001-054375 A2	26.07.2001
		WO 2001-053940 A3	26.07.2001
		WO 2001-053940 A2	26.07.2001
		WO 2000-005637 A3	03.02.2000
		WO 2000-005637 A2	03.02.2000
		WO 2002-027556 A1	04.04.2002
US 06393437 B1	21.05.2002	None	
US 2005-0207732 A1	22.09.2005	CN 100454999 C	21.01.2009
		CN 1674664 A	28.09.2005
		EP 1580754 A1	28.09.2005
		KR 20050094285 A	27.09.2005
		KR 10-0585668 B1	07.06.2006