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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 March 2012 (01.03.2012)

PCT

(10) International Publication Number
WO 2012/024800 A1

(51) International Patent Classification:
G06F 17/30 (2006.01) *G06F 12/16* (2006.01)

(21) International Application Number:
PCT/CA2011/050514

(22) International Filing Date:
24 August 2011 (24.08.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/376,905 25 August 2010 (25.08.2010) US

(72) Inventors; and
(71) Applicants : **ZACHARIASSEN, Rayan** [DK/CA]; 89 Kingsway Crescent, Toronto, Ontario M8X 2R8 (CA). **LAMB, Steven** [CA/CA]; 15 Ballacaine Drive, Toronto, Ontario M8Y 4A7 (CA).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **FERNANDES, Laryn-Joe** [CA/CA]; 41 Harkins Drive, Ajax, Ontario L1T 3V1 (CA).

(74) Agents: **MCMILLAN LLP** et al.; Brookfield Place, Suite 4400, 181 Bay Street, Toronto, Ontario M5J 2T3 (CA).

(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, QA, 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, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, 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**: METHOD AND SYSTEM FOR EXTENDING DATA STORAGE SYSTEM FUNCTIONS

(57) **Abstract**: A method and system for extending the functionality of a data storage system, the data storage system including a data organization means and a data storage resource, the method including the steps of dissociating the data storage functions of the data storage system from other functions of the data storage system and transferring at least a portion of said data storage functions to an intercepting system.

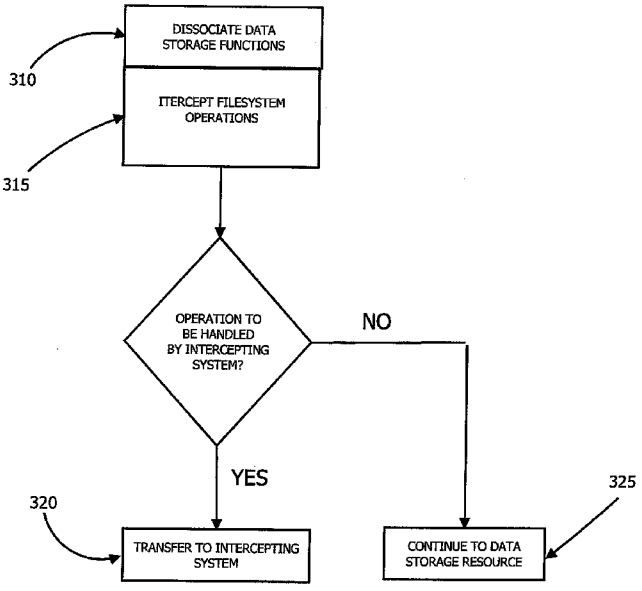

FIGURE 3

WO 2012/024800 A1

METHOD AND SYSTEM FOR EXTENDING DATA STORAGE SYSTEM FUNCTIONS

[0001] This application claims priority from United States Provisional Application No. 61/376,905, filed on August 25, 2010, the contents of which are incorporated herein in their entirety by reference.

Field of the Invention

[0002] The present invention relates generally to data storage systems, and more particularly, to a method and system for extending the functions of a data storage system.

Background of the Invention

[0003] A filesystem on a computer enables the operating system of the computer to act as a trusted third party that enforces security and naming protocols between communicating processes, even where such processes are not active at the same time. One of the functions of a filesystem is to catalogue and organize data provided to it so that it can later be retrieved. In order to carry out this function, filesystems must manage a storage resource. Typical storage resources appear to the filesystem as a contiguous byte sequence, a contiguous block sequence, or essentially a key-value store for example in an object storage system.

[0004] Filesystems are typically part of the operating system on a computer, but may also exist as an extension of the operating system in an application, or as a pure application accessed over a network connection using a client filesystem protocol. Various examples of these are known in the art. In all cases a filesystem is designed to use a particular kind of storage resource, for example a disk, and will have constraints related to that storage resource. These constraints could be related to, for example, performance, capacity, parallelism, scalability, physical location, etc. Additional constraints exist within the filesystem itself in terms of its features allowing manipulation of the data it is storing, for example to apply encryption, compression, replication, or other transformations including context sensitive processing (data polymorphism) when the filesystem is not provided with this functionality in advance. In the art, the phrase context sensitive generally refers to a

program feature that changes depending on what you are doing in the program. For example, context sensitive help provides documentation for the particular feature that you are in the process of using and context or context sensitive processing of data allows data to be processed differently depending on how or where the data will be used.

[0005] Prior art systems and methods of coping with some of these constraints has been to use virtual devices that create a virtual storage resource for the filesystem to use. Due to the lack of available context that exists within the filesystem and is not passed on to the storage resource, this approach is unable to provide context sensitive processing of the data and is also unable to remove constraints that are due to a 1:1 mapping of data the filesystem sees to data actually on the storage resource, for example the total data storage capacity.

[0006] It is therefore an object of the invention to provide a novel system and method for extending the functions of a data storage system, for example to extend the functions of a data storage system to permit context sensitive data processing.

Summary of the Invention

[0007] According to one embodiment of the invention, there is provided a method for extending the functionality of a data storage system, the data storage system including a data organization means and a data storage resource, the method including dissociating the data storage functions of the data storage system from other functions of the data storage system and transferring at least a portion of the data storage functions to an intercepting system.

[0008] According to one aspect of the invention, the intercepting system is a complimentary storage resource. The data organization means may be selected from the group comprising a filesystem, a key-value store and a database.

[0009] According to another aspect of the invention, the dissociating step is carried out by providing an interceptor means in communication with the data organization means, the data storage resource and the intercepting system; the interceptor means intercepting a filesystem operation to

determine whether a function of the operation should be handled by the data storage resource or by the intercepting system.

[0010] According to another aspect of the invention, the interceptor means intercepts a filesystem operation while the operation still has context and before the operation would otherwise be decomposed into independent operations suitable for the data storage resource.

[0011] According to another aspect of the invention, at least another portion of the data storage functions are retained with the data storage system. According to another aspect of the invention, all of the data storage functions are transferred to the intercepting system.

[0012] According to another aspect of the invention, the data storage system is selected from the group comprising a filesystem, a key-value store, an object store and a network protocol.

[0013] According to another aspect of the invention, the data storage system is shared among multiple external interfaces.

[0014] According to another aspect of the invention, the interceptor means comprises a user-space application program in cooperation with a facility of an operating system or of a filesystem. According to another aspect of the invention, the interceptor means comprises a filesystem protocol proxy application performed on a network. According to another aspect of the invention, the interceptor means comprises a minifilter driver adapted to intercept filesystem operations in an operating system kernel.

[0015] According to another aspect of the invention, the intercepting system uses one or more complimentary storage resources to carry out its functions, the complimentary storage resources being independent from the storage resource.

[0016] According to another aspect of the invention, the intercepting system implements capacity expansion of the data storage system. According to another aspect of the invention, the intercepting system improves the performance of the data storage system by altering one or more

characteristics of data on the storage resource. The one or more characteristics are preferably selected from the group comprising a storage format, a storage location and storage order.

[0017] According to another aspect of the invention, the method further includes the step of carrying out de-duplication by the intercepting system. According to another aspect of the invention, the method further includes the step of carrying out data polymorphism by the intercepting system. According to another aspect of the invention, the method for includes the step of step of implementing independent access control mechanisms for the data by the intercepting system. According to another aspect of the invention, the method for includes the step of versioning data by the intercepting system.

[0018] According to another aspect of the invention, the method further includes the step of implementing one of single or multi-level caching of the data storage system; wherein the implementing step is carried out by the interceptor means. According to another aspect of the invention, the method further includes the step of implementing locality optimization by pushing less-used data to remote data storage systems and by pulling more-used data to nearby data storage systems; wherein the implementing step is carried out by the interceptor means. The remote data storage systems include data storage systems that are either physically remote or require more time to access.

[0019] According to another aspect of the invention, the method further includes the step of implementing name based virtualization making file names that are not valid in a current data storage system appear to be valid by referring to data on other data storage systems; wherein the implementing step is carried out by the interceptor means.

[0020] According to another aspect of the invention, the method further includes the step of implementing data backup and data replication; wherein the implementing step is carried out by the interceptor means.

[0021] According to another aspect of the invention, the method further includes the step of implementing data virtualization including allowing data under the same name to be physically

located in different data storage systems; wherein the implementing step is carried out by the interceptor means.

[0022] According to another aspect of the invention, the method further includes providing distinct capabilities for selected data at the interception system than at the storage resource, wherein the distinct capabilities are selected from the group comprising performance characteristics, de-duplication, data polymorphism, independent access control mechanisms, versioning, caching, locality, replication and data virtualization. The selected data is preferably identified using a selection mechanism employing a metadata pattern matching of one or more selected from the group comprising name, timestamps, size, historical information, physical information and contextual information.

[0023] According to another embodiment of the invention, there is provided a system for extending the functionality of a data storage system, the data storage system including a data organization means and a data storage resource, the system comprising a dissociating means for dissociating data storage functions of the data storage system from other functions of the data storage system and a means for transferring at least a portion of the data storage functions to an intercepting system.

[0024] According to one aspect of this embodiment, the intercepting system may be a complimentary storage resource.

[0025] According to another aspect of this embodiment, the dissociating means comprises an interceptor in communication with the data organization means, the data storage resource and the intercepting system; the interceptor adapted to intercept a filesystem operation to determine whether a function of the operation should be handled by the data storage resource or by the intercepting system.

[0026] According to another aspect of this embodiment, the interceptor intercepts a filesystem operation while the operation still has context and before the operation would otherwise be decomposed into independent operations suitable for the data storage resource.

[0027] According to another aspect of this embodiment, at least another portion of the data storage functions are retained with the data storage system. Alternatively, all of the data storage functions are transferred to the intercepting system.

[0028] According to another aspect of this embodiment, the data storage system is selected from the group comprising a filesystem, a key-value store, an object store and a network protocol.

[0029] According to another aspect of this embodiment, the data storage system is shared among multiple external interfaces.

[0030] According to another aspect of this embodiment, the interceptor comprises a user-space application program in cooperation with a facility of an operating system or of a filesystem.

[0031] According to another aspect of this embodiment, the interceptor comprises a filesystem protocol proxy application performed on a network.

[0032] According to another aspect of this embodiment, the interceptor comprises a minifilter driver adapted to intercept filesystem operations in an operating system kernel.

[0033] According to another aspect of this embodiment, the intercepting system uses one or more complimentary storage resources to carry out its functions, the complimentary storage resources being independent from the storage resource.

[0034] According to another aspect of this embodiment, the intercepting system implements capacity expansion of the data storage system.

[0035] According to another aspect of this embodiment, the intercepting system improves the performance of the data storage system by altering one or more characteristics of data on the storage resource.

[0036] According to another aspect of this embodiment, the one or more characteristics are selected from the group comprising a storage format, a storage location and storage order.

[0037] According to another aspect of this embodiment, the intercepting system is adapted to perform de-duplication of the data.

[0038] According to another aspect of this embodiment, the intercepting system is adapted to perform polymorphism of the data.

[0039] According to another aspect of this embodiment, the intercepting system is adapted to perform independent access control mechanisms for the data.

[0040] According to another aspect of this embodiment, the intercepting system is adapted to perform versioning of the data.

[0041] According to another aspect of this embodiment, the interceptor is adapted to perform one of single or multi-level caching of the data storage system.

[0042] According to another aspect of this embodiment, the interceptor is adapted to perform locality optimization by pushing less-used data to remote data storage systems and by pulling more-used data to nearby data storage systems.

[0043] According to another aspect of this embodiment, the remote data storage systems include data storage systems that are either physically remote or require more time to access.

[0044] According to another aspect of this embodiment, the interceptor is adapted to perform name based virtualization making file names that are not valid in a current data storage system appear to be valid by referring to data on other data storage systems.

[0045] According to another aspect of this embodiment, the interceptor is adapted to perform one of data backup and data replication.

[0046] According to another aspect of this embodiment, the interceptor is adapted to perform data virtualization including allowing data under the same name to be physically located in different data storage systems.

[0047] According to another aspect of this embodiment, the interception system is adapted to perform one or more selected from the group comprising improving performance characteristics, de-duplication, data polymorphism, independent access control mechanisms, versioning, caching, locality, replication and data virtualization.

[0048] According to another aspect of this embodiment, the data organization means is selected from the group comprising a filesystem, a key-value store and a database.

Brief Description of the Drawings

[0049] Embodiments will now be described, by way of example only, with reference to the attached Figures, wherein:

[0050] Figure 1 shows a high-level architecture of a system according to the invention.

[0051] Figure 2 shows a computer system in which the invention may be used and/or implemented.

[0052] Figure 3 shows an embodiment of the method according to the invention.

Detailed Description of the Embodiments

[0053] The invention provides a novel system and method for affecting all data related constraints, including but not limited to those that can be affected using virtual storage resources. The invention is able to provide this functionality for data storage systems other than filesystems that provide higher level abstractions above the actual storage resources as their primary interface,

for example databases and particular object databases, key value stores, certain network protocols and certain shared data systems.

[0054] Furthermore, the invention allows for all the traditional constraints of data storage systems to be changed without the cooperation of, or changes made to, the filesystem itself. Thus, the invention does not require a complete overhaul or implementation of a new filesystem and permits for the advantages and extended functions described herein to be implemented on existing systems, in the interfaces between older and newer systems and in developing new systems as well with or without the need for a new filesystem to be developed. Broadly, the invention provides for an intercepting system to be attached to the filesystem that can selectively intercept filesystem operations while they still include context information before these operations are decomposed into operations suitable for the storage resource. That is, the filesystem includes contextual operations and instructions for carrying them out that are provided by the operating system of the computer. When the filesystem interacts with the storage resource, these contextual operations and instructions are lost as the filesystem-storage resource interaction is only concerned with the retrieval, storage, and cataloguing of data. The intercepting system according to the invention provides the functionality the filesystem requires at an interception point where the intercepting system's operation is transparent to the filesystem. Thus, the invention provides for a method and system that extends the functionality of data storage resources and systems by dissociating the storage responsibility of the data storage system from its other functions (such as file naming, locking, sharing, security, etc.), and optionally having the actual storage responsibility being carried out by a separate component.

[0055] With reference to Figure 1, there is shown one embodiment of the invention in which there is shown a data storage system 100, an interceptor 200 and an intercepting system 300. Details of the preferred embodiments in which various distributions of prior art data storage system functions are dissociated from the data storage system itself and divided up to be carried out by either the data storage system or the intercepting system are described below. The novel dissociation and subsequent distribution of functionality between the data storage system 100 and the intercepting system 300 is believed to be novel in the art, and particularly with respect to providing a distribution of functionality between the data storage system 100 and the intercepting

system 300 in a non-cooperative environment. However, although the preferred embodiment of the invention relates to an add-on mechanism to existing non-cooperative data storage systems, it can also be an architectural feature providing extensibility to a cooperative data storage system to provide extended and enhanced functionality, examples of which will be described in more detail below.

[0056] With reference to Figure 3, there is shown another embodiment of the invention, where there is provided a method for extending the functionality of a data storage system including the steps of dissociating the data storage functions 310 of a data storage system from other functions of the data storage system by intercepting 315 a filesystem operation to determine whether a function of the operation should be handled by the data storage resource of by an intercepting system and transferring 320 at least a portion of the data storage functions to the intercepting system. Alternatively, the intercepted data storage functions continue to the data storage resource 325.

[0057] The invention generally operates within the context of a computer system, and serves to provide an extension of the data storage capabilities associated with a general computer system, an exemplary one of which is shown in Figure 2. As shown, the computer system 20 has a number of physical and logical components, including a central processing unit ("CPU") 24, random access memory ("RAM") 28, an input/output ("I/O") interface 32, a network interface 36, non-volatile storage 40, and a local bus 44 enabling the CPU 24 to communicate with the other components. The CPU 24 executes an operating system and a number of software systems. RAM 28 provides relatively-responsive volatile storage to the CPU 24. The I/O interface 32 allows for input to be received from one or more devices, such as a keyboard, a mouse, etc., and outputs information to output devices, such as a display and/or speakers. The network interface 36 permits communication with other systems. Non-volatile storage 40 stores the operating system and programs. During operation of the computer system 20, the operating system, the programs and the data may be retrieved from the non-volatile storage 40 and placed in RAM 28 to facilitate execution.

[0058] Data storage system 100 is generally known in the art, and for the purposes of the invention is any machine, device or apparatus and is able to store data using a given identifier (such

as a filename), and later retrieve at least a portion of it, on demand by that identifier. Data storage system 100 includes a data organization means 105 and uses an operationally independent, but optionally physically embedded, machine or module referred to herein as storage resource 110 (also shown as non-volatile storage 40 in Figure 2) to store data in a primitive form according to the characteristics of the storage resource, from where the data storage system itself can later retrieve the stored data. Data storage system 100 may be shared among multiple external interfaces.

[0059] Representative examples of a data organization means 105 include filesystems, key-value stores, databases, and other machines layered on top of these such as web caches and page files, in combination with . Representative examples of storage resources 110 are physical disks, memory such as RAM, RAID arrays, paper tape, documents, etc. In a representative example of the preferred embodiment, the storage system 100 may be a Windows™ NTFS filesystem and the storage resource may be a hard disk drive.

[0060] Interceptor means 200 is a system for intercepting and processing data operations, such as filesystem operations. According to the invention, the interceptor means 200 intercepts a data operation while the operation still has context available but before the operation is decomposed into the appropriate context free and more specific operations for the storage resource, as is the case for virtually all storage resources. Context may be, for example, application information including the source of the data request, details of the data being requested, details of what the data will be used for and other information requiring knowledge of the current data operation.

[0061] According to the representative example discussed above, the interceptor means 200 is a minifilter or legacy filter driver designed and otherwise arranged to intercept filesystem operations at the appropriate level in the Windows™ operating system kernel. Such a driver may be independently implemented or it may be procured commercially. In other examples, the interception mechanism could be provided by facilities built into other operating environments, for example, but not limited to filesystem stacks, using application level filesystem providers like FUSE (filesystem in user space), STREAMS drivers for network intercepts, or physical transparent proxy machines. The interceptor means may also be a custom designed software module incorporating the

functionality herein described. Interceptor means 200, referred to interchangeable as an interceptor, may also be a filesystem protocol proxy application.

[0062] The interceptor means 200 manages the junction between the data storage system 100 and the intercepting system 300. Intercepting system 300 has an interest in certain operations of the data storage system. In the representative example, this could be all data inputs/outputs in order to redirect the data to another storage device. The intercepting system 300 would configure the intercepting mechanism to pass all data input/output operations laterally to the intercepting system 300, instead of passing them through to the lower layers of the data storage system 100, as is traditionally done.

[0063] Upon receipt of an intercepted operation, the intercepting system 300 has a number of options: (a) the intercepting system 300 could determine that it does not process this operation and instruct the interceptor means 200 to carry on as if there was no intercept, so that the interceptor means passes the operation on to the lower levels of the data storage system 100; (b) the intercepting system 300 could determine that it needs to modify the operation before passing the operation on to the lower levels of the data storage system 100; or, (c) the intercepting system 300 could determine that it is responsible for handling the operation itself and take over responsibility for it, in which case the interception mechanism 200 must react to the upper and lower levels of the data storage system appropriately for that operation, including all error handling.

[0064] The intercepting system 300 may be any number of hardware devices, such as a complimentary storage resource, a multimedia extender, a multimedia server, a home server, an application specific controller, or any similar system that is capable of performing various tasks, including those that are typically performed by storage resources.

[0065] In accordance with the invention, there is provided the interceptor means 200, by which the intercepting system is able to transparently, that is in a manner not identifiable by the data storage system, take responsibility for intercepted operations. This allows an existing filesystem to be extended with new functionality when the data storage resource is dissociated from the filesystem. In contrast, prior art intermediary programs that perform some data interception

functions, such as virus checkers and encryption shims, use the original data storage system to provide their functionality.

[0066] The invention further provides an existing data storage system to separate the location of stored data or metadata from the data storage system while retaining the original semantics of that data storage system. Thus, the data storage system remains responsible for any existing data or metadata already stored, but the intercepting system can take over responsibility for any new or modified data or metadata according to its own policy as long as the external semantics of the existing data storage system remain unchanged.

[0067] In the representative example described throughout, the original filesystem remains responsible for all metadata, and the intercepting system takes over responsibility for some or all data. Specifically, the invention allows the NTFS filesystem metadata, including directories, the Master File Table (MFT), and the contents of each MFT entry, to be maintained on the filesystem's storage resources as would normally occur, but the data itself could be manipulated separately and stored independently of the configuration of the original data storage system.

[0068] The invention provides a number of advantages, including providing a way for a legacy, unmodified NTFS implementation to support modern filesystem features like replication, de-duplication, caching, pooling, as well as features usually provided by storage resources (at RAID levels), and new unique features. The invention makes this possible in a manner that is compatible with existing applications because unmodified semantics are presented at the external application interface of the filesystem. That is, the filesystem remains unchanged and is useable by the operating system without modification.

[0069] One prior art example of attempting to solve a similar problem as solved by the invention, is the Microsoft™ Windows Home Server project (WHS), a data storage system with simple expandability intended for home users. WHS initially contained a function called Drive Extender that extended NTFS on the Windows system by using a file based paradigm, i.e. whole files could be placed on a different destination NTFS filesystem and a reference (a "tombstone") left on the original NTFS filesystem. This approach failed because it was not completely transparent to

applications (the reference that was left behaved differently than the original file), and in fact the functionality Drive Extender was trying to provide was initially re-implemented using a virtual block storage resource model, that is by appearing as a storage resource to the filesystem. Subsequently the functionality was completely dropped from WHS. Other examples of the original Drive Extender approach are apparent in the art, for example in Hierarchical Storage Management systems. The present invention, on the other hand, allows the filesystem semantics and metadata to remain with the original storage resource, and be referred to as needed, but the data itself is manipulated and stored independently without the constraints of the configuration of the original data storage system.

[0070] While the example of the preceding paragraph has been provided with respect to disk drive storage systems, the approach of the invention may also be used with respect to extending a non-cooperative filesystem to other non-cooperative data storage systems, such as key-value stores, object stores, various kinds of caches, databases, etc. Given the ability to locate data separately, differently, or even just with additional processing compared to the design of the original data storage system, it is now possible to provide a number of new features and corresponding functionality to applications using an existing data storage system. Applicant believes that the prior art has not disclosed a method and system for dissociating the data storage responsibility from the other functions of a data storage system to thereby extend the functionality of the data storage system beyond the constraints of the filesystem and the data storage system itself.

[0071] One of the advantages of incorporating the system and method according to the invention is a possible increase in performance. Redirected data can be stored on a faster storage resource than the original data storage system uses, or it could be stored in a way which leads to faster reading and/or writing of the data, or both mechanisms could be used simultaneously. For example, the metadata of the storage system can be maintained on the original storage resource, but the data itself stored on the faster storage resource. In this example, the interceptor means directs the filesystem to retrieve metadata from the original storage resource, but to retrieve data from the faster storage resource. When the information and data are then passed on to the filesystem for processing by the operating system, the data appears as it would have if it were retrieved directly from the original storage resource but has been retrieved much faster. Performance may also be

improved, for example, by caching data on a secondary storage resource that is possibly faster or has other improved characteristics.

[0072] In one embodiment of the invention the redirected data is stored in sparse files on the faster storage resource. A sparse file is a file that contains holes where data is not allocated on the storage resource but represented virtually. One of the problems in file caching systems is to represent the cached file in a space-efficient format, usually necessitating the creation of a data packing module to store actual data, and to translate between internal file location and where the data for those locations is stored. In order to cache data at a sub-file level as needed, the data that has not yet been retrieved from the original file should be represented virtually, not physically. Using sparse files allows any filesystem supporting sparse files to provide this functionality, and saves having to develop and provide this functionality within the invention. Applicant believes that the use of sparse files to cache the data of real files is novel, and a person skilled in the art will appreciate the ability to use existing capabilities to provide this internal function of a sub-file cache.

[0073] The invention may also be used in the data compression context, where data could be compressed in various ways that are not limited by the characteristics of the original data storage system. For example, normal stream compression or de-duplication compression where the data is not stored again if it already exists somewhere in the system. This type of data compression is typically not possible directly at the level of the data storage system using the teachings of the prior art. A person skilled in the art will appreciate the distinct advantages of providing data compression within any filesystem and data storage resource system, irrespective of the filesystem and data storage resource system being used.

[0074] Another possible application is in data polymorphism where data can appear different dependent on the context in which it is intended to be used. While various data polymorphism applications are known in the art, these are typically performed at the application level and is constrained by the data storage system. According to the method of the invention, data can be manipulated at the interception system, that is at the data storage system level and be presented to the filesystem in a context-modified manner so operating system and application resources or modification is not required.

[0075] In the context of a document management system, the invention allows for redirected data to be versioned in a manner which retains the data storage system semantics, but provides an extended semantics that allows retrieval of older versions of the data. Current systems require older versions of the data either to be updated to incorporate the extended semantics, or to be retrieved in a manner that differentiates older data from newer data.

[0076] Redirected data inputs/outputs could also provide input to a distributed data storage system that controls where the source data is located based on its time related usefulness to the data storage system. Again, in this regard, these features can be implemented at the data storage system level and will not appear any differently to the application accessing the filesystem.

[0077] In the name virtualization context, data may no longer reside on the original data storage system's storage resource, and could be located somewhere completely different without have to provide adaptations to the filesystem. Data virtualization becomes possible too, where a single piece of named data may no longer reside on the same storage resource, but could be scattered to multiple data storage systems and/or storage resources. In the data replication context, redirected data inputs/outputs could drive a replication process to ensure redundancy of data.

[0078] While data and operations on data as have been described above have known implementations, mainly on the application level, such as by using a virtual machine implementation, the disadvantages of these implementations are described in the background of invention section, the invention allows for these features and functionality to be provided at the data storage system level, and therefore makes them independent of the filesystem or the operating system.

[0079] Various adaptations and implementations of the invention may be made without departing from the spirit of the claims that follow, including implementing, by the interceptor, one of single or multi-level caching of the data storage system; wherein the implementing step is carried out by the interceptor means and implementing locality optimization by pushing less-used data to remote data storage systems and by pulling more-used data to nearby data storage systems. The

remote data storage systems include data storage systems that are either physically remote or require more time to access. Furthermore, it is possible to implement, by the interceptor, name based virtualization making file names that are not valid in a current data storage system appear to be valid by referring to data on other data storage systems. Data backup and replication is also possible by having the interceptor communicate directly with the filesystem. Data virtualization is also made possible at the data storage system level, including allowing data under the same name to be physically located in different data storage systems; wherein the implementing step is carried out by the interceptor means.

[0080] According to another aspect of the invention, the method further includes providing distinct capabilities for selected data at the interception system than at the storage resource, wherein the distinct capabilities are selected from the group comprising performance characteristics, de-duplication, data polymorphism, independent access control mechanisms, versioning, caching, locality, replication and data virtualization. The selected data is preferably identified using a selection mechanism employing a metadata pattern matching of one or more selected from the group comprising name, timestamps, size, historical information, physical information and contextual information.

[0081] The above-described embodiments are intended to be examples of the present invention and alterations and modifications may be effected thereto, by those of skill in the art, without departing from the scope of the invention that is defined solely by the claims appended hereto.

What is claimed is:

1. A method for extending the functionality of a data storage system, said data storage system including a data organization means and a data storage resource, the method comprising dissociating the data storage functions of the data storage system from other functions of the data storage system and transferring at least a portion of said data storage functions to an intercepting system.

2. A method according to claim 1, wherein said intercepting system comprises a complimentary storage resource.

3. A method according to claim 1, wherein said dissociating step is carried out by providing an interceptor means in communication with the data organization means, the data storage resource and the intercepting system; said interceptor means intercepting a filesystem operation to determine whether a function of said operation should be handled by said data storage resource or by said intercepting system.

4. A method according to claim 3, wherein said interceptor means intercepts a filesystem operation while the operation still has context and before the operation would otherwise be decomposed into independent operations suitable for the data storage resource.

5. A method according to claim 1, wherein at least another portion of said data storage functions are retained with said data storage system.

6. A method according to claim 1, wherein all of said data storage functions are transferred to said intercepting system.

7. A method according to claim 1, wherein said data storage system is selected from the group comprising a filesystem, a key-value store, an object store and a network protocol.

8. A method according to claim 1, wherein the data storage system is shared among multiple external interfaces.

9. A method according to claim 3, wherein said interceptor means comprises a user-space application program in cooperation with a facility of an operating system or of a filesystem.

10. A method according to claim 3, wherein said interceptor means comprises a filesystem protocol proxy application performed on a network.

11. A method according to claim 3, wherein said interceptor means comprises a minifilter driver adapted to intercept filesystem operations in an operating system kernel.

12. A method according to claim 3, wherein said intercepting system uses one or more complimentary storage resources to carry out its functions, said complimentary storage resources being independent from said storage resource.

13. A method according to claim 1, wherein said intercepting system implements capacity expansion of the data storage system.

14. A method according to claim 1, wherein said intercepting system improves the performance of the data storage system by altering one or more characteristics of data on the storage resource or by using a complimentary storage resource.

15. A method according to claim 14, wherein said one or more characteristics are selected from the group comprising a storage format, a storage location and storage order.

16. A method according to claim 1, further comprising the step of carrying out de-duplication by said intercepting system.

17. A method according to claim 1, further comprising the step of carrying out data polymorphism by said intercepting system.

18. A method according to claim 1, further comprising the step of implementing independent access control mechanisms for the data by said intercepting system.

19. A method according to claim 1, further comprising the step of versioning data by said intercepting system.

20. A method according to claim 12, further comprising the step of implementing one of single or multi-level caching of the data storage system; wherein said implementing step is carried out by said interceptor means.

21. A method according to claim 12, wherein the complimentary storage resource is the operating system memory cache normally used to cache data.

22. A method according to claim 12, further comprising the step of implementing locality optimization by pushing less-used data to remote data storage systems and by pulling more-used data to nearby storage resources or data storage systems; wherein said implementing step is carried out by said interceptor means.

23. A method according to claim 21, wherein said remote data storage systems include data storage systems that are either physically remote or require more time to access.

24. A method according to claim 12, further comprising the step of implementing name based virtualization making file names that are not valid in a current data storage system appear to be valid by referring to data on other data storage systems; wherein said implementing step is carried out by said interceptor means.

25. A method according to claim 12, further comprising the step of implementing one of data backup and data replication; wherein said implementing step is carried out by said interceptor means.

26. A method according to claim 12, further comprising the step of implementing data virtualization including allowing data under the same name to be physically

located in different data storage systems; wherein said implementing step is carried out by said interceptor means.

27. A method according to claim to claim 1, further comprising providing distinct capabilities for selected data at said interception system than at said storage resource, wherein said distinct capabilities are selected from the group comprising performance characteristics, de-duplication, data polymorphism, independent access control mechanisms, versioning, caching, locality, replication and data virtualization.

28. A method according to claim 26, wherein said selected data is identified using a selection mechanism employing a metadata pattern matching of one or more selected from the group comprising name, timestamps, size, historical information, physical information and contextual information.

29. A method according to claim 1, wherein said data organization means is selected from the group comprising a filesystem, a key-value store and a database.

30. A method according to claim 12, wherein said complementary storage resource is a sparse file.

31. A system for extending the functionality of a data storage system, said data storage system including a data organization means and a data storage resource, the system comprising a dissociating means for dissociating data storage functions of the data storage system from other functions of the data storage system and a means for transferring at least a portion of said data storage functions to an intercepting system.

32. A system according to claim 31, wherein said intercepting system comprises a complimentary storage resource.

33. A system according to claim 31, wherein said dissociating means comprises an interceptor in communication with the data organization means, the data storage

resource and the intercepting system; said interceptor adapted to intercept a filesystem operation to determine whether a function of said operation should be handled by said data storage resource or by said intercepting system.

34. A system according to claim 33, wherein said interceptor intercepts a filesystem operation while the operation still has context and before the operation would otherwise be decomposed into independent operations suitable for the data storage resource.

35. A system according to claim 31, wherein at least another portion of said data storage functions are retained with said data storage system.

36. A system according to claim 31, wherein all of said data storage functions are transferred to said intercepting system.

37. A system according to claim 31, wherein said data storage system is selected from the group comprising a filesystem, a key-value store, an object store and a network protocol.

38. A system according to claim 31, wherein the data storage system is shared among multiple external interfaces.

39. A system according to claim 33, wherein said interceptor comprises a user-space application program in cooperation with a facility of an operating system or of a filesystem.

40. A system according to claim 33, wherein said interceptor comprises a filesystem protocol proxy application performed on a network.

41. A system according to claim 33, wherein said interceptor comprises a minifilter or legacy filter driver adapted to intercept filesystem operations in an operating system kernel.

42. A system according to claim 33, wherein said intercepting system uses one or more complimentary storage resources to carry out its functions, said complimentary storage resources being independent from said storage resource.

43. A system according to claim 33, wherein said intercepting system implements capacity expansion of the data storage system.

44. A system according to claim 33, wherein said intercepting system improves the performance of the data storage system by altering one or more characteristics of data on the storage resource.

45. A system according to claim 33, wherein said one or more characteristics are selected from the group comprising a storage format, a storage location and storage order.

46. A system according to claim 31, wherein said intercepting system is adapted to perform de-duplication of the data.

47. A system according to claim 31, wherein said intercepting system is adapted to perform polymorphism of the data.

48. A system according to claim 31, wherein said intercepting system is adapted to perform independent access control mechanisms for the data.

49. A system according to claim 31, wherein said intercepting system is adapted to perform versioning of the data.

50. A system according to claim 33, wherein said interceptor is adapted to perform one of single or multi-level caching of the data storage system.

51. A system according to claim 33, wherein the complimentary storage resource is the operating system memory cache normally used to cache data.

52. A system according to claim 33, wherein said interceptor is adapted to perform locality optimization by pushing less-used data to remote data storage systems

and by pulling more-used data to nearby storage resources or data storage systems.

53. A system according to claim 51, wherein said remote data storage systems include data storage systems that are either physically remote or require more time to access.

54. A system according to claim 33, wherein said interceptor is adapted to perform name based virtualization making file names that are not valid in a current data storage system appear to be valid by referring to data on other data storage systems.

55. A system according to claim 33, wherein said interceptor is adapted to perform one of data backup and data replication.

56. A system according to claim 33, wherein said interceptor is adapted to perform data virtualization including allowing data under the same name to be physically located in different data storage systems.

57. A system according to claim to claim 31, wherein said interception system is adapted to perform one or more selected from the group comprising improving performance characteristics, de-duplication, data polymorphism, independent access control mechanisms, versioning, caching, locality, replication and data virtualization.

58. A system according to claim 31, wherein said data organization means is selected from the group comprising a filesystem, a key-value store and a database.

59. A system according to claim 42, wherein said complementary storage resource is a sparse file.

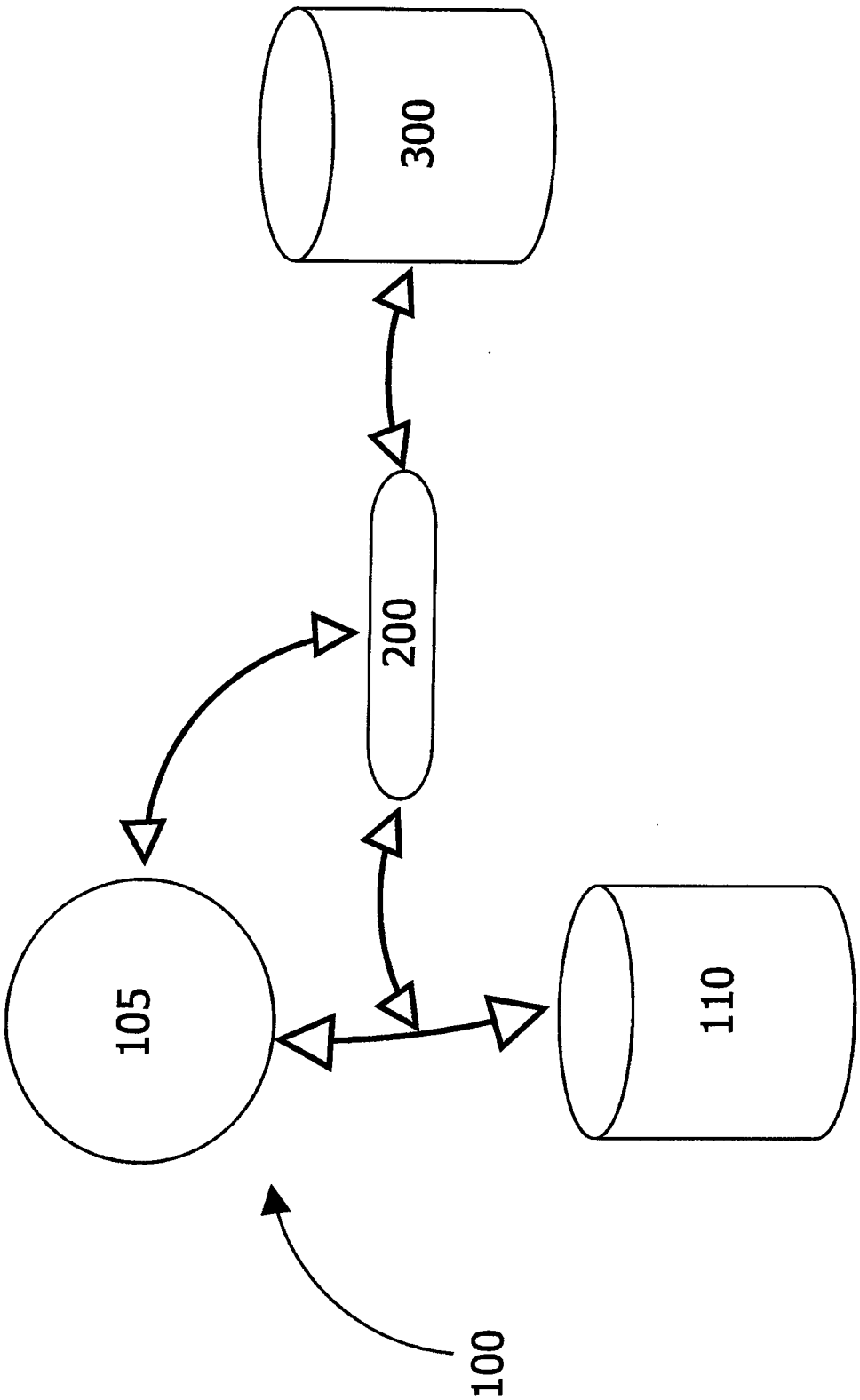

FIGURE 1

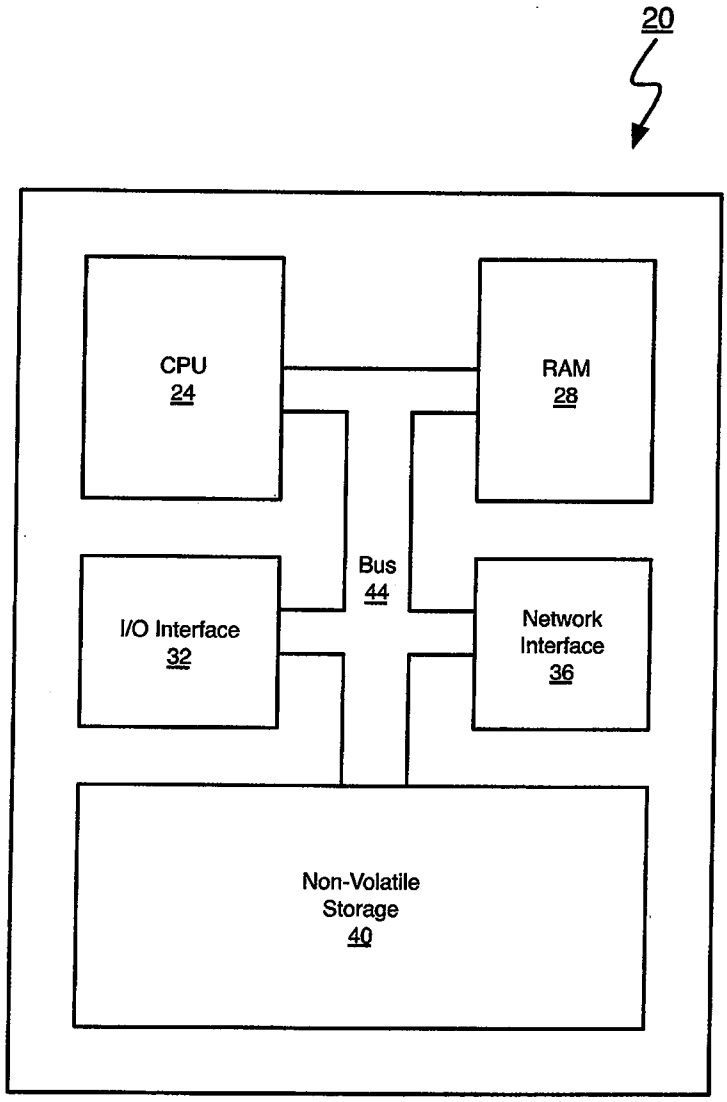
20
CPU
24
RAM
28
Bus
44
I/O Interface
32
Network
Interface
36
Non-Volatile
Storage
40

FIGURE 2

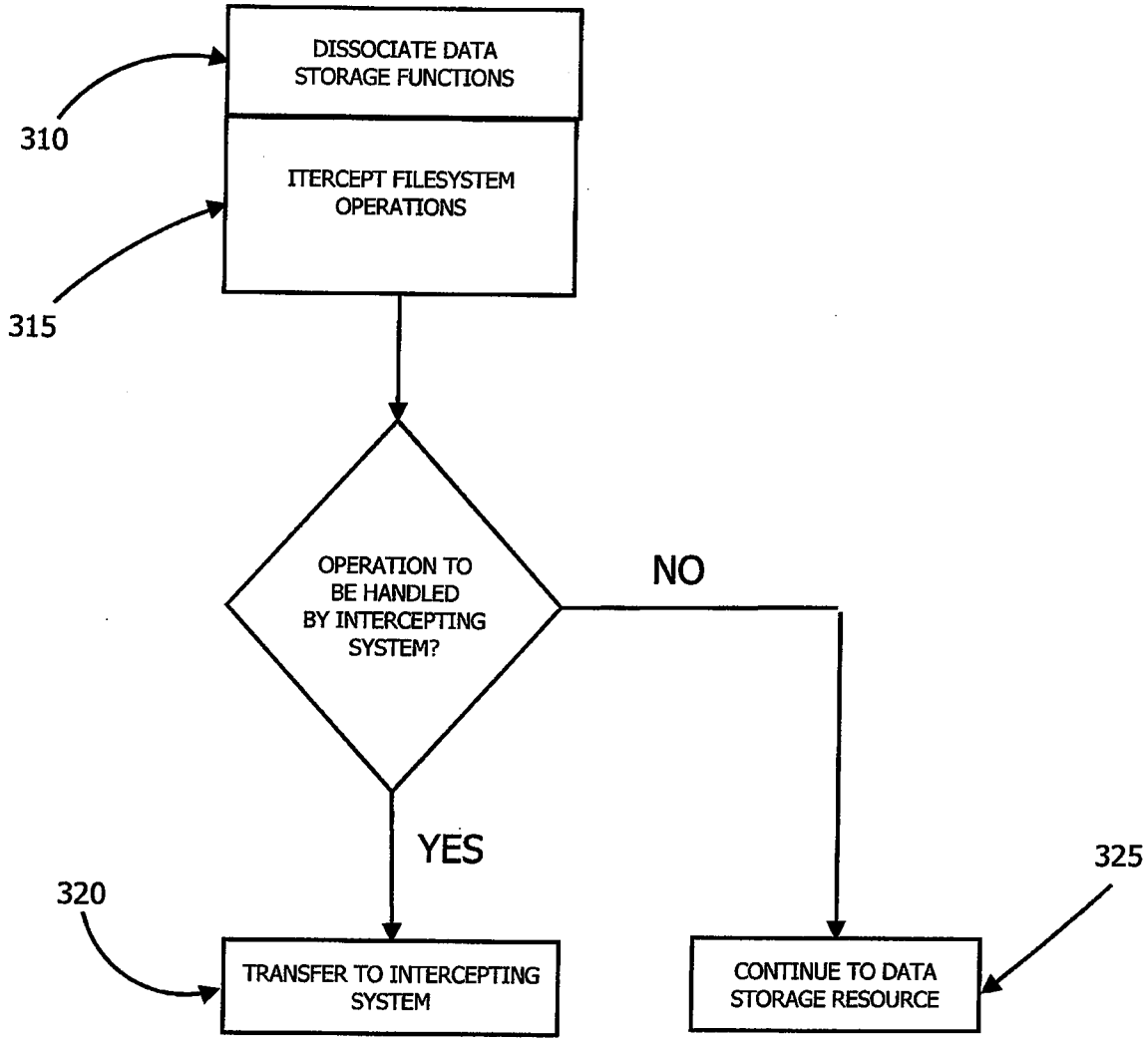
DISSOCIATE DATA STORAGE FUNCTIONS
310
ITERCEPT FILESYSTEM OPERATIONS
315
OPERATION TO BE HANDLED BY INTERCEPTING SYSTEM?
NO
YES
320
TRANSFER TO INTERCEPTING SYSTEM
325
CONTINUE TO DATA STORAGE RESOURCE

FIGURE 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2011/050514

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***G06F 17/30*** (2006.01) , ***G06F 12/16*** (2006.01)
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)
IPC (2006.01): ***G06F 17/30***, ***G06F 12/16***

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

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
Databases: TotalPatents and EPOQUE (Epodoc and txten).
Keywords: storage, system, dissociating, file, kernel, intercept, hierarchy, organization, function, sparse, separating, database.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP0856803A2 (BALABINE ET AL.) 05 August 1998 (05-08-1998) *Whole Document*	1-59
A	CA2646776A1 (SEDLAR ET AL.) 15 February 2001 (15-02-2001) *Whole Document*	
A	WO2010/037117A1 (BYERS ET AL.) 1 April 2010 (01-04-2010) *Whole Document*	

[ ] 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 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	Date of mailing of the international search report
07 October 2011 (07-10-2011)	30 November 2011 (30-11-2011)

Name and mailing address of the ISA/CA	Authorized officer
Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	Tony Khoury (819) 934-7882

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
EP0856803A2	05 August 1998 (05-08-1998)	AU739236B2	04 October 2001 (04-10-2001)
		AU5275798A	06 August 1998 (06-08-1998)
		BR9800065A	02 March 1999 (02-03-1999)
		CA2228210A1	31 July 1998 (31-07-1998)
		CA2228210C	02 September 2008 (02-09-2008)
		DE69839744D1	04 September 2008 (04-09-2008)
		EP0856803A3	10 January 2001 (10-01-2001)
		EP0856803B1	23 July 2008 (23-07-2008)
		JP10247155A	14 September 1998 (14-09-1998)
		US5937406A	10 August 1999 (10-08-1999)
		US6442548B1	27 August 2002 (27-08-2002)
CA2646776A1	15 February 2001 (15-02-2001)	AU762942B2	10 July 2003 (10-07-2003)
		AU774090B2	17 June 2004 (17-06-2004)
		AU3596700A	04 September 2000 (04-09-2000)
		AU6495400A	05 March 2001 (05-03-2001)
		AU2002334706B2	22 November 2007 (22-11-2007)
		AU2002334721B2	23 October 2008 (23-10-2008)
		AU2002334747B2	30 October 2008 (30-10-2008)
		AU2003287565A1	03 June 2004 (03-06-2004)
		AU2003287565A2	07 July 2005 (07-07-2005)
		AU2003287565B2	09 November 2006 (09-11-2006)
		AU2003287565C1	31 July 2008 (31-07-2008)
		AU2003290654A1	03 June 2004 (03-06-2004)
		AU2003290654B2	27 August 2009 (27-08-2009)
		AU2003290655A1	03 June 2004 (03-06-2004)
		AU2003290655B2	25 September 2008 (25-09-2008)
		AU2004203240A1	02 September 2004 (02-09-2004)
		AU2004203240B2	03 April 2008 (03-04-2008)
		AU2004203241A1	02 September 2004 (02-09-2004)
		AU2004203241B2	10 April 2008 (10-04-2008)
		AU2004203242A1	02 September 2004 (02-09-2004)
		AU2004203242B2	01 May 2008 (01-05-2008)
		AU2004203243A1	02 September 2004 (02-09-2004)
		AU2004203243B2	20 March 2008 (20-03-2008)
		AU2004203249A1	02 September 2004 (02-09-2004)
		AU2004203249B2	03 April 2008 (03-04-2008)
		CA2359880A1	24 August 2000 (24-08-2000)
		CA2359880C	04 July 2006 (04-07-2006)
		CA2379930A1	15 February 2001 (15-02-2001)
		CA2461854A1	10 April 2003 (10-04-2003)
		CA2461854C	23 November 2010 (23-11-2010)
		CA2461871A1	10 April 2003 (10-04-2003)
		CA2462300A1	03 April 2003 (03-04-2003)
		CA2504141A1	27 May 2004 (27-05-2004)
		CA2504141C	16 June 2009 (16-06-2009)
		CA2505156A1	27 May 2004 (27-05-2004)
		CA2505156C	05 July 2011 (05-07-2011)
		CA2505158A1	27 May 2004 (27-05-2004)
		CA2505158C	04 May 2010 (04-05-2010)
		CA2650251A1	15 February 2001 (15-02-2001)
		CN1561496A	05 January 2005 (05-01-2005)
		CN1295636C	17 January 2007 (17-01-2007)
		CN1561497A	05 January 2005 (05-01-2005)
		CN1299223C	07 February 2007 (07-02-2007)
		CN1585945A	23 February 2005 (23-02-2005)
		CN1585945B	18 May 2011 (18-05-2011)
		CN1711534A	21 December 2005 (21-12-2005)
		CN100432993C	12 November 2008 (12-11-2008)
		CN1717656A	04 January 2006 (04-01-2006)
		CN100351791C	28 November 2007 (28-11-2007)
		CN1729467A	01 February 2006 (01-02-2006)
		CN100429654C	29 October 2008 (29-10-2008)
		DE60310255D1	18 January 2007 (18-01-2007)
		DE60310255T2	28 June 2007 (28-06-2007)
		DE60325758D1	26 February 2009 (26-02-2009)

Continued on page 4...

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
		EP1145143A2	17 October 2001 (17-10-2001)
		EP1330727A2	30 July 2003 (30-07-2003)
		EP1433089A2	30 June 2004 (30-06-2004)
		EP1440394A2	28 July 2004 (28-07-2004)
		EP1446737A2	18 August 2004 (18-08-2004)
		EP1559006A2	03 August 2005 (03-08-2005)
		EP1559035A2	03 August 2005 (03-08-2005)
		EP1559035B1	06 December 2006 (06-12-2006)
		EP1559036A2	03 August 2005 (03-08-2005)
		EP1559036B1	07 January 2009 (07-01-2009)
		EP1852790A2	07 November 2007 (07-11-2007)
		EP1852790A3	27 February 2008 (27-02-2008)
		EP1876542A2	09 January 2008 (09-01-2008)
		EP1876542A3	30 July 2008 (30-07-2008)
		EP1898321A2	12 March 2008 (12-03-2008)
		EP1898321A3	08 October 2008 (08-10-2008)
		HK1077107A1	19 June 2009 (19-06-2009)
		HK1077108A1	23 March 2007 (23-03-2007)
		JP2005505059A	17 February 2005 (17-02-2005)
		JP4351530B2	28 October 2009 (28-10-2009)
		JP2006505872A	16 February 2006 (16-02-2006)
		JP4406609B2	03 February 2010 (03-02-2010)
		JP2005505042A	17 February 2005 (17-02-2005)
		JP4443221B2	31 March 2010 (31-03-2010)
		JP2006505871A	16 February 2006 (16-02-2006)
		JP4476813B2	09 June 2010 (09-06-2010)
		JP2003505748U	12 February 2003 (12-02-2003)
		JP2003527659A	16 September 2003 (16-09-2003)
		JP2005505058A	17 February 2005 (17-02-2005)
		JP2006505877A	16 February 2006 (16-02-2006)
		JP2010152916A	08 July 2010 (08-07-2010)
		US6427123B1	30 July 2002 (30-07-2002)
		US6549916B1	15 April 2003 (15-04-2003)
		US2003037056A1	20 February 2003 (20-02-2003)
		US6571231B2	27 May 2003 (27-05-2003)
		US6922708B1	26 July 2005 (26-07-2005)
		US2004088306A1	06 May 2004 (06-05-2004)
		US6947950B2	20 September 2005 (20-09-2005)
		US6950822B1	27 September 2005 (27-09-2005)
		US6965903B1	15 November 2005 (15-11-2005)
		US2004088340A1	06 May 2004 (06-05-2004)
		US7020653B2	28 March 2006 (28-03-2006)
		US7028037B1	11 April 2006 (11-04-2006)
		US7047250B1	16 May 2006 (16-05-2006)
		US7047253B1	16 May 2006 (16-05-2006)
		US2003065659A1	03 April 2003 (03-04-2003)
		US7051033B2	23 May 2006 (23-05-2006)
		US7051039B1	23 May 2006 (23-05-2006)
		US7092967B1	15 August 2006 (15-08-2006)
		US2003140308A1	24 July 2003 (24-07-2003)
		US7096224B2	22 August 2006 (22-08-2006)
		US2004064466A1	01 April 2004 (01-04-2004)
		US7120645B2	10 October 2006 (10-10-2006)
		US2006101041A1	11 May 2006 (11-05-2006)
		US7158981B2	02 January 2007 (02-01-2007)
		US7280995B1	09 October 2007 (09-10-2007)
		US2004088415A1	06 May 2004 (06-05-2004)
		US7308474B2	11 December 2007 (11-12-2007)
		US2003033285A1	13 February 2003 (13-02-2003)
		US7366708B2	29 April 2008 (29-04-2008)
		US2005065949A1	24 March 2005 (24-03-2005)
		US7386568B2	10 June 2008 (10-06-2008)
		US7418435B1	26 August 2008 (26-08-2008)
		US2005091287A1	28 April 2005 (28-04-2005)
		US7502782B2	10 March 2009 (10-03-2009)
		US7620620B1	17 November 2009 (17-11-2009)
		US2008215528A1	04 September 2008 (04-09-2008)

Continued on page 5...

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
		WO0049533A2	24 August 2000 (24-08-2000)
		WO0049533A3	05 July 2001 (05-07-2001)
		WO0111486A2	15 February 2001 (15-02-2001)
		WO0111486A3	17 April 2003 (17-04-2003)
		WO03027908A2	03 April 2003 (03-04-2003)
		WO03027908A3	12 February 2004 (12-02-2004)
		WO03030031A2	10 April 2003 (10-04-2003)
		WO03030031A3	12 February 2004 (12-02-2004)
		WO03030032A2	10 April 2003 (10-04-2003)
		WO03030032A3	29 April 2004 (29-04-2004)
		WO2004044738A2	27 May 2004 (27-05-2004)
		WO2004044738A3	24 February 2005 (24-02-2005)
		WO2004044780A2	27 May 2004 (27-05-2004)
		WO2004044780A3	09 December 2004 (09-12-2004)
		WO2004044781A2	27 May 2004 (27-05-2004)
		WO2004044781A3	20 January 2005 (20-01-2005)
WO2010/037117A1	01 April 2010 (01-04-2010)	US2010198889A1	05 August 2010 (05-08-2010)