A computer-implemented method for providing storage resources (R01-R09) of a storage management system (201), the method comprising: receiving (101) a storage service request comprising an indication of a service class (301-303); determining (102) all resource managers (RM1-RM5) mapped to the indicated service class according to a first mapping (212); each resource manager having a generic API; for each of the determined resource managers, determine (104) selected ones of all the storage resources controllable by said resource manager; gathering (105) monitoring data of the selected storage resources; comparing (106) the gathered monitoring data for calculating a score for each one of the selected storage resources; and calling (107) at least one function of the resource manager operable to control the storage resource having an optimum score.

Figure 2

Title of the Invention: Providing storage resources upon receipt of a storage service request

Abstract Title: System providing storage as a service
Receiving a storage service request comprising an indication of a service class

Determining all resource managers mapped to the indicated service class according to a first mapping

For each of the determined resource managers, execute:

Determine selected ones of all the storage resources controllable by said resource manager

Gathering monitoring data of the selected storage resources

Comparing the gathered monitoring data with the requirements of the indicated service class for calculating a score for each selected storage resource

Calling at least one function of the resource manager operable to control the storage resource having the optimum score, thereby configuring or initializing said storage resource to provide the requested storage service

Figure 1
DESCRIPTION

Providing Storage Resources upon Receipt of a Storage Service Request

Field of the invention

The invention relates to the field of data storage management, and more particularly to the provision of storage resources as a service.

Background

The demand for the automated management of an IT infrastructure, in particular the management and provision of storage space has been constantly growing. The provision of storage space to a plurality of clients is typically based on a contract comprising a so-called "service level agreement" (SLA) specifying the level of service a storage provider has agreed to provide to the respective client. Service level agreements can contain numerous storage capacity and/or storage performance metrics which may be expressed in the form of service-level objectives (SLOs). Technically, storage capacity may be provided by various forms of IT services, e.g. by means of a cloud computing environment or other service based delivery models. An example of a cloud server environment being operable to perform data storage operations supporting a variety of clients that may connect of the system is disclosed in WO 2011/002777 A2. Another example of a storage infrastructure is described in US 2009/0182777 A1. The described storage infrastructure ensures that customer data is stored on appropriate storage components satisfying a set of SLO policies.

Current IT-systems used by service providers to offer and provide storage capacity to their clients are inflexible. In case the service provider offers multiple different service levels corresponding to different sets of SLOs to the clients, each of said service levels is tightly coupled to a particular set of storage resources being operable to provide their respective storage capacity in accordance with their respectively assigned service level and SLOs. For example, hard discs or storage pools with very short response times (which tend to be expensive) may be assigned to a
high-quality (and more expensive) service level while slow storage resources, e.g. tape drives, may be assigned to a service level suited for doing back-up jobs. Said assignment is based on a direct mapping of parameters required by the respective proprietary Application Programming Interfaces (APIs) of the storage resources to the service level provided. Often, said mapping is specified within compiled code sections of the storage management system. Adding or removing storage resources to the storage management system or reassigning a storage resource to another service level requires a manual modification and maybe even recompilation of elements of the service management infrastructure. Due to the plurality of storage models, storage types, storage vendors and other factors from which an API providing access to a given storage resource may depend, the configuration and assignment of the storage resources to different service levels is a time consuming task in state of the art storage management systems, resulting in inflexible systems being difficult to maintain and operate.

Some other state of the art systems allow the clients to directly query and access the storage resources. However, said approach does not increase the flexibility of storage resource assignment, but merely shifts the problems associated with maintaining an association of service levels to proprietary storage resource APIs from the service provider to the client.

The inflexible assignment of storage resources to service levels further bears the problem that multiple requests for the same storage service might cause bottlenecks and available storage capacity may not be optimally used.

**Summary of the invention**

It is an objective of embodiments of the invention to provide for an improved computer implemented method, computer-readable medium and computer system for providing storage resources. The objective is solved by the features of the independent claims. Preferred embodiments are given in the dependent claims. If not explicitly indicated otherwise, embodiments of the invention can be freely combined with each other.
A 'storage management system' (SMS) as used herein encompasses any kind of monolithic or distributed program logic being operable to receive storage service requests from one or more client devices and being operable to automatically provide storage capacity to the clients in response to said request. The program logic may be software-based, firmware-based and/or hardware-based. The storage capacity is derived from one or more storage resources being controlled by the SMS. Controlling a storage resource may comprise configuring, initializing and/or managing of storage capacities and functions provided by a storage resource in order to provide storage capacities of said storage resources as the requested service. For example, the IBM Tivoli storage productivity center may be used as the SMS.

A 'storage resource' as used herein encompasses one or more computer-readable storage devices or storage device components which are controlled in a concerted manner via a common interface. Typically, said interface is specific to the vendor and/or the type and/or the model of the storage resource. A storage resource may be any of the following: a physical storage volume, a logical storage volume, a physical storage device comprising one or more physical storage volumes, or a storage pool comprising a plurality of physical and/or logical storage volumes. A physical storage volume may be a hard disk, a hard disk partition, or a Logical Unit Number (LUN) of a storage device. The SMS may treat any one of the physical volumes as a sequence of chunks called physical extents. A logical storage volume is a storage volume whose storage blocks (or 'blocks') are mapped to one or more physical storage volumes. A physical storage device can be, for example, a Fiber Channel disk, a Mirrored disk, a RAID protected disk, a SATA drive, a network attached storage, a short tape, a WORM tape, an Encryption Tape, a Long tape, a Flash drive, solid state discs (SSD), Serial ATA discs (SATA) and various optical storage devices. A storage resource of a SMS may be physically integrated within the computer system hosting the SMS or may be connected to said computer system via a network connection, typically a TCP/IP connection.
A ‘service level objective’ (SLO) as used herein is a set of one or more parameter-value pairs being descriptive of e.g. the available storage capacity, storage accessibility, retention time, the security of the stored data or performance. Said parameter-value pairs may e.g. indicate the mean time between failures, mean time to repair, data transfer rates or similar measurable details being characteristic for a particular service level. At least some of the service level objectives may be used to specify the minimum or maximum property value(s) allowable for any storage capacity provided as a service of a selected service level having assigned said SLOs.

A ‘service class’ as used herein is a data object representing a predefined service level of a storage service, whereby said service level corresponds to a set of requirements any storage service provided in accordance with said service level has to comply to. Preferentially, said requirements are represented in the form of one or more service level objectives (SLOs) corresponding to said service level and being stored in association with the service class representing said service level. A data object may be, for example, an entry in a relational database, a file, in particular an XML file, or a binary file. It may also be an element of a file, e.g. an XML element of an XML file or a line in a file comprising a table.

A ‘resource manager’ as used herein is a runtime data object exposing a generic API to a SMS for allowing said SMS to control one or more storage resources controlled by said resource manager via said generic API. A runtime data object is a program logic executed by a processor. A resource manager may comprise one or more program routines executed upon receipt of a call via the generic API, wherein said program routines are operable to call proprietary methods of a respectively controlled storage resource via a proprietary interface of said storage resource.

A ‘generic API’ of a resource manager as used herein is an application program interface (API) allowing to access and call some functions of the resource manager via said generic API. The term ‘generic’ denotes that the API is not specific for the vendor or model of the storage resources con-
controlled by the resource manager. According to preferred embodiments, the
generic API comprises methods and attributes being specified in accordance
with a non-proprietary meta-model of storage functions and capabilities to
be provided by the SMS to a client upon request, whereby the generic API
abstracts from the particularities of the storage resources. Said particularities may relate to the vendor, manufacturer or other property of the
storage resource.

The term 'monitoring data' indicates a computational load of a storage re-
source at a given moment in time. Monitoring data may comprise, for exam-
ple, the occupied and/or free storage space on a storage resource, the num-
ber of input and/or output operations (I/O operations) per time executed on
said storage resource, the amount of data written to and/or read from the
storage resource, a response time in respect to an I/O request, etc.

A 'storage service request' as used herein is a computer-interpretable re-
quest of a client, e.g. a mobile phone or a computer system, for being
granted access to storage capacity of a particular kind and offering a par-
ticular level of service.

A 'mapping' as used herein is a storage space comprising data assigning two
or more entities to each other. The assignment may be implemented e.g. by
one or more tables of a relational database or by one or more files, e.g.
XML files.

A 'storage model' as used herein is an identifier of the physical device
type providing or underlying the storage resource. The storage model is a
manufacturer- and/or vendor coined category. Examples of two storage models
provided by a vendor "IBM" are "IBM DS8000" and "IBM XIV".

A 'storage type' as used herein denominates the type of access to data
stored on a storage resource supported by said storage resource. For exam-
ple, a storage resource having organized its stored data within a file sys-
tem may have the storage type 'file system', another storage resource ac-
cessible via block-based SCSI commands may have the storage type ‘block’ while the data of still another storage resource may be accessible via a proprietary data I/O interface such as Amazon’s S3 protocol, said storage resource having the storage type ‘object’. A ‘vendor’ as used herein is an identifier of the organization having sold the storage resource and/or having manufactured the storage resource.

In one aspect, the invention relates to a computer-implemented method for providing storage resources of a SMS. The method comprises:

- receiving a storage service request comprising an indication of a service class, a service class being a data object having assigned one or more requirements (SLOs) to be fulfilled by any storage service provided in accordance with said service class;

- determining all resource managers mapped to the indicated service class according to a first mapping, each resource manager being a runtime data object comprising a generic API providing control over one or more of the storage resources;

- for each of the determined resource managers, determine selected ones of all the storage resources controllable by said resource manager;

- gathering monitoring data of the selected storage resources, the monitoring data being indicative of free capacities of the selected storage resources;

- comparing the gathered monitoring data with the requirements of the indicated service class for calculating a score for each one of the selected storage resources; depending on the implementation, the amount of the score value may positively or negatively correlate with the amount and/or quality of free storage capacity of said storage resource; and

- calling at least one function of the resource manager operable to control the storage resource having an optimum score via said resource manager’s generic API, thereby configuring or initializing said storage resource to provide the requested storage service in accordance with the indicated service class.
Said features may be advantageous as a highly flexible coupling of the requested service level (represented by the indicated service class) and the storage resources actually used for providing the requested storage capacity is provided. The coupling allows a dynamic (at runtime of the SMS) selection of the storage resources used for providing the service in dependence on the current load of the storage resources. Thus, a better use of available storage capacity is provided and costs for additional hardware can be avoided.

The usage of resource managers as an additional instance between the individual storage resources and the service classes is advantageous as said additional mapping element increases the flexibility of the mapping and allows mapping service classes to generic, storage-related functions provided by the resource managers rather than directly mapping the service classes to proprietary interfaces of the storage resources. This feature may also ease maintainability, because the only components which are tightly bound to the proprietary interfaces of the storage resources are the storage managers. As the resource managers expose a generic PI, the first mapping may allow to flexibly and generically re-assign service classes to different abstract storage functions (such as formatting, indexing, mirroring, etc) provided by other resource managers without burdening the operator with the requirement to dig into the details of any proprietary API of one of the storage resources.

According to preferred embodiments, the first mapping (between service classes and resource managers) is a data structure that can be edited and becomes effective without any recompilation of source code. The first mapping may be, for example, a configuration file or an entry in a relational database. Said features may be advantageous as the service provider may reassign a plurality of storage resources to a different service level by editing a single entry comprising a mapping of the resource manager controlling said resources to another service level.
The storage request may be a request for storage capacity in accordance with the SLOs assigned to the requested service class. For example, a storage service request for backup storage space will typically indicate a service class requiring storage space with large storage capacity, whereby comparatively large response times may be considered as acceptable. A storage service request for storage to be used by an application program executing a plurality of I/O operations will typically require the assignment of storage resources having short response time. According to some embodiments, the storage capacity is provided in response to the service request by means of a cloud computing environment. Upon receipt of the storage service request, the SMS may invoke each of the selected resource managers, i.e., resource managers associated with resources able to provide the requested service class, in order to obtain a score for resources managed by said the selected resource managers respectively. This score may be used by the SMS in deciding which resources to allocate storage capacity on in order to fulfill said request. Preferentially, the SMS selects a scorecalculation algorithm from a plurality of available score calculation algorithms in dependence on the indicated service class. In dependence on the selected algorithm, different input parameters may be used such as the rate of input/output operations per time unit (“I/O rate”), or the I/O rate per capacity unit, or simply the amount of unused storage capacity on the resource. Preferentially, providing storage capacity is implemented as granting the client read and/or write access to storage capacity provided by the one or more storage resources having assigned an optimum score value. Said features may result in a form of load balancing across resources available to fulfill a storage service request, e.g. a request for a particular amount of storage capacity, thereby preventing bottlenecks.

In case of a positive correlation of score value and the amount of free storage capacity and/or the level of storage quality, the highest calculated score value is considered as optimum score. Otherwise, the lowest calculated score value is considered the optimum score. The score may be indicative of free storage capacities in respect to the requirements of all available service classes or selectively in respect to the indicated ser-
vice class. According to some embodiments, the score is calculated for all storage resources of the SMS by the same scoring algorithm taking as input, for example, the free storage space, the current I/O load etc. According to other embodiments, the score is calculated by a scoring algorithm being particular to the service class indicated in the request. This may be advantageous as a storage service request for backup storage capacity may have different requirements (SLOs) than, for example, a request for storage space to be used by an I/O intensive application program (see above).

According to embodiments, upon adding a new storage resource to the SMS, the new storage resource a) being of a new vendor of the storage resource, and/or b) being of a new storage type, and/or c) being of a new storage model, and/or d) offering a new function, adds a new resource manager to the SMS. Said adding may comprise updating the first mapping for assigning the new resource manager to one or more service classes. The new resource manager comprises new executable routines, whereby said routines: a) are vendor-specific, and/or b) are storage type specific, and/or c) are storage model specific, and/or d) implement the new function. Thereby, the resource management system is enabled to configure or initialize said new storage resource for providing the new storage resource upon request, in accordance with the service class the resource manager is assigned to.

According to embodiments, for each unique combination of two, three or four out of a set of properties of the storage resources, providing a resource manager implementing executable routines particular to said unique combination of properties, said properties comprising: the vendor, and/or the storage type, and/or the storage model, and/or a function of said storage resource. Said features may be advantageous as a highly fine-grained control on a plurality of different storage resources is provided by means of a plurality of resource managers respectively comprising program routines being particular to the vendor, type, model and/or a function of the new storage resource.
According to embodiments, the generic APIs of all storage managers operable to control storage resources which share a function share a corresponding method interface. Said method interface enables the SMS to control to said shared function generically. New functions of new storage resources are exposed to the SMS by means of the resource manager’s generic API. Thus, the resource management system is enabled to call functions shared by multiple resource managers via a shared generic method interface of the generic API of the respective resource managers without being affected by the proprietary details of the added storage resources. A ‘shared method interface’ shared by multiple APIs as used herein is a method interface having the same method signature in all APIs. The implementation of said method in the respective resource managers may differ.

According to embodiments, the generic APIs of all storage managers having assigned storage resources sharing a storage type share one or more storage-type-specific method interfaces. The storage-type-specific method interfaces enable the SMS to control one or more respective storage-type-specific functions.

According to embodiments, the determination of the selected ones of all the storage resources controllable by any one of the determined resource managers comprises: accessing the first mapping, wherein each pair of mapped service class and resource manager is flagged by a first parameter value; in case said first parameter value of any one of said pairs has a first data value (e.g. ‘yes’), using all storage resources controllable by the resource manager of said pair as the selected storage resources of said resource manager; in case said first parameter value of the given pair has a second data value (e.g. ‘no’), not using any of the storage resources controllable by the resource manager of said pair as the selected storage resource of said resource manager.

According to embodiments, the determination of the selected ones of all the storage resources controllable by one of the determined resource managers comprises:
- accessing a second mapping, the second mapping comprising pairs of service classes and storage resources mapped to each other, wherein each of said pairs is flagged by a second parameter value;

- in case said second parameter value of any one of said pairs has a third data value (e.g. 'ASSIGNED'), using the storage resource of said pair as one of the selected storage resources;

- in case said second parameter value of said pair has a fourth data value (e.g. 'UNASSIGNED'), not using the storage resource of said pair as the selected storage resources.

Flagging the first and/or second mapping with a first and/or second parameter value may be advantageous as said kind of mapping minimizes the time and effort required by an operator for specifying the mapping.

In case a plurality of storage resources are controlled by a given resource manager and shall almost all be used to provide storage capacity in case of a request for said service class, it is easier to map all storage resources controllable by said resource manager per default to the service class via a 'yes' data value in the first mapping and then exclude some few storage resources from being used for said service class by adding an "UNASSIGNED" data value in the second mapping for each of said few storage resources.

In case a plurality of storage resources is controlled by a given resource manager and only a few of them shall be used to provide storage capacity in case of a request for said service class, it is easier to un-map all storage resources controllable by said resource manager per default to the service class via a 'no' data value in the first mapping and then explicitly map said few storage resources to said service class by adding an "ASSIGNED" data value in the second mapping for each of said few storage resources.

According to embodiments, the selected ones of the storage resources are determined in accordance with the first and the second mapping, wherein the mapping information in the second mapping for determining the selected
storage resources overrule the respective mapping information in the first mapping.

According to embodiments, the method further comprises: generating a list of service classes supported by the SMS; (optionally): for each service class in the list, provide the monitoring data gathered and aggregated from all storage resources currently assigned via their resource managers to said service class; thus, the capacity utilization of all storage resources currently mapped to said service class may be displayed; providing interface means for outputting the list of service classes (and optionally also for outputting the monitoring data having been aggregated on a per-service-class basis) and for allowing a requestor of the storage service to select one of the service classes from the list and to include an indication of the selected service class into the storage service request submitted via said interface.

The list of service classes may, for example, be generated by an operator of the SMS who specifies one or more service classes by editing a file for assigning one or more LSOs any service provided in accordance with said service class is required to comply to. Outputting the list of service classes and/or monitoring data may comprise generating, by the storage management module, a user interface window, e.g. a HyperText Markup Language (HTML) page, and providing this window to the clients e.g. via the Internet. The HTML page may comprise a HyperText Transfer Protocol (HTTP) GET or POST form allowing the selection of a service class and an automated submission of a storage service request comprising an indication of the selected class to the SMS.

According to embodiments, the method further comprises adding a new service class associated with new requirements to the SMS. In case one or more of the resource managers are already operable to control a storage resource for providing said storage resource in accordance with the new requirements, the method comprises mapping the new service class to said one or more resource managers in the first mapping. This is sufficient for providing storage capacity in accordance with the new service class. In case none
of the resource managers is operable to control a storage resource for providing said storage resource in accordance with the new requirements, the method comprises creating a new resource manager operable to control at least one of the storage resources for providing said at least one storage resource in accordance with said new requirements and mapping the new resource manager to the new service class in the first mapping.

Said features may be advantageous e.g. in the following use case scenario: a first storage resource of a given type, model and vendor and a corresponding resource manager operable to control said first storage resource are already part of the SMS. Then, a second storage resource of the same model, type and vendor as the first one is added to the system, whereby the configuration of the second storage resource differs from that of the first one. For example, the second storage resource may be configured to delete its stored data after an expiration time specified in the configuration of said storage resource. In this case, it is not necessary to create an additional resource manager as the existing one can also control the second storage resource. Nevertheless, the functionalities provided by the new configuration can easily and without the need to rewrite and recompile any code be propagated to the clients by specifying a new service class. In the described use case scenario, a service class may be specified having assigned some SLOs which require that all data stored on a storage provided in accordance with said service class are deleted after the predefined expiration date, thus providing a secure but temporary storage service. Preferentially, the first storage resource not comprising said special configuration is not assigned to said new service class, e.g. by means of further mappings as is in the following described for some embodiments.

According to embodiments, the adding of the new service class is executed upon adding a new storage function to the SMS. The new function may be added by adding a new storage resource, by modifying and extending a configuration of an existing storage resource or by adding a new storage resource controllable by an existing storage manager, whereby the new storage resource comprises a new configuration. The new requirements are require-
ments in respect to said new storage function. As explained for the use case scenario above, a new function such as the deletion of data after an expiration time may correspond to a set of new requirements (SLOs) that data stored to a storage resource provided in accordance with said requirements must be deleted after said expiration time.

According to embodiments, the method further comprises, for each one of all service classes supported by the SMS:

- selecting a current one of the service classes;
- determining the resource managers assigned to the current service class according to the first mapping;
- determining selected ones of all the storage resources controllable by said determined resource managers; in particular, said selected ones should be operable to provide their storage capacity in accordance with the requirements of the current service class. As explained in the use case scenario above, a given resource manager may be operable to control a first and a second storage resource of the same storage type and model but having different configurations and thus possibly being in compliance with different sets of requirements.
- gathering monitoring data of the selected storage resources for providing monitoring data of the current service-class;
- assembling the monitoring data of all service classes for outputting a report.

Thus, the report may be indicative of occupied capacities of all storage resources adapted for providing a storage service in accordance with any one of said service classes’ requirements. Said features may be advantageous as they may allow a client to select a service class in dependence on said service class’s current load and/or may allow an operator of the SMS to determine any service classes having continuously assigned too many or too few storage resources.

According to embodiments, the gathering of monitoring data of the selected storage resources is executed by one of the resource managers being operable to control said selected storage resources and being mapped according
to the first mapping to the service class indicated in the service request. Alternatively, said service class may be the current service class in a loop executed for determining the available storage capacity for all service classes. Said features may be advantageous as they allow implementing distributed system architectures wherein the resource managers are hosted on other processing devices than e.g. the computer system evaluating the first and/or second mapping. Thus, the performance of the system may be increased. According to embodiments, the score calculation is executed by the selected storage resources. The SMS may select a score calculation algorithm in dependence on the requested service class and send an identifier of the selected storage algorithm to all selected resource managers, thereby allowing said resource managers to calculate the scores for their respectively controlled resources via said indicated algorithm and to return the calculated score to the SMS. Thus, it is ensured that the score is calculated by different resource managers in the same manner. In addition, the calculation of the score can be parallelized and distributed among different resource managers to increase performance.

In a further aspect, the invention relates to a computer-readable medium comprising computer-readable program code embodied therewith, which, when executed by a processor, cause the processor to execute a method according to anyone of the embodiments specified above.

In a further aspect the invention relates to a computer system operatively coupled to a plurality of storage resources R01-R09. The computer system comprises an interface for receiving a storage service request and a SMS for managing the storage resources.

The SMS is adapted for:

- receiving the storage service request via the interface, the storage service request comprising an indication of a service class, a service class being a data object having assigned one or more requirements (SLOs) to be fulfilled by any storage service provided in accordance with said service class;
- determining all resource managers mapped to the indicated service class according to a first mapping, each resource manager being a runtime data object comprising a generic API providing control over one or more of the storage resources;

- for each of the determined resource managers, determine selected ones of all the storage resources controllable by said resource manager;

- gathering monitoring data of all selected storage resources, the monitoring data being indicative of free capacities of the selected storage resources;

- comparing the gathered monitoring data with the requirements of the indicated service class for calculating a score for each one of the selected storage resources; and

- calling at least one function of the resource manager operable to control the storage resource having an optimum score via said resource manager’s generic API, thereby configuring or initializing said storage resource to provide the requested storage service in accordance with the indicated service class.

**Brief Description of the Drawings**

In the following, preferred embodiments of the invention will be described in greater detail by way of example, only making reference to the drawings in which:

Fig. 1 shows a flowchart of an embodiment of the invention,

Fig. 2 shows a SDS and three clients,

Fig. 3 is an overview on a plurality of mappings, and

Fig. 4 illustrates an assignment of multiple service classes, resource managers and storage resources.

**Detailed description**

As will be appreciated by one skilled in the art, aspects of the present invention may be embodied as a system, method or computer program product. Accordingly, aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining
software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, aspects of the present invention may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon. Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

A cloud computing environment is service oriented with a focus on statelessness, low coupling, modularity, and semantic interoperability. At the heart of cloud computing is an infrastructure comprising a network of interconnected cloud computing nodes. It is understood in advance that although this disclosure may be used in a cloud computing environment, implementation of the teachings recited herein are not limited to a cloud computing environment. Rather, embodiments of the present invention are capable of being implemented in conjunction with any other type of computing environment now known or later developed. Cloud computing is a model of service delivery for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, network bandwidth, servers, processing, memory, storage, applications, virtual machines, and services) that can be rapidly provisioned and released with
minimal management effort or interaction with a provider of the service. This cloud model may include at least five characteristics and at least three service models. Characteristics may be as follows: On-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. Service Models may be as follows: Software as a Service (SaaS), platform as a service (PaaS), Infrastructure as a Service (IaaS). In the following, like numbered elements in these figures are either similar elements or perform an equivalent function.

**Figure 1** shows a flowchart of a method for providing storage resources of a SMS as depicted in figure 2. In the following, the method will be described by making reference to elements of figures 1 and 2. In a first step 101, the SMS 201 hosted by computer system 200 receives a storage service request which comprises an identifier of a service class. The request may be submitted by an application program 206 of a client computer system 205 via a network 207. Each service class may have assigned a set of requirements, also referred herein as service level objectives SLOs 210 any storage provided in accordance with said service class needs to comply to. The SMS determines in step 102 all resource managers RM1-RM4 which are mapped to the service class indicated in the request according to a first mapping 212. In step 103 the SMS begins to loop over the determined resource managers for executing the steps 104-107 for each of the determined resource managers. In step 104, the SMS determines selected ones of all the storage resources R01-R09 controllable by the current resource manager of the loop. The totality of storage resources controllable by the current resource manager may be hard coded within the source code of the resource manager or maybe implemented by a mapping referred herein as 'general mapping' 209 which may be, for example, a file or a database entry. The determination of the selected ones may be based on a parameter value contained in the first mapping 212 and optionally also on mapping information contained in the second mapping 211. In step 105, the storage management module gathers monitoring data from the selected storage resources. The gathering may be implemented based on a push or pull approach. According to embodiments, each resource manager is operable to monitor all storage resources the re-
source manager is in control of and to forward the monitoring data to a control module 213 of the SMS 201. The SMS compares in step 106 the gathered monitoring data with the requirements 210 of the indicated service class for calculating a score for each selected storage resource. After having calculated a score for each selected storage resource for each of the determined resource managers, the resource management system calls in step 107 at least one function of the resource manager operable to control the storage resource having an optimum score. Calling said resource manager comprises configuring or initializing said storage resource for providing the requested storage service to the client in accordance with the selected service class’s requirements. For example, a client may send a request for storage capacity of 500 GB being indicative of a service class ‘Retention-Controlled’ to the SMS. The SMS may interpret said request as a request to create a storage capacity of type ‘file’ with capacity 500 GB, and allocate said storage capacity to resources being operable to provide ‘RetentionControlled’ storage space and being not already occupied or under heavy load. The response may consist merely of a path the client may use to access the requested storage capacity. In addition or alternatively, the response may include a confirmation that the requested capacity value (like here, “500 GB”) is available and can be accessed. The response may also comprise some credentials for accessing the storage capacity.

Figure 2 shows a block diagram of the SMS 201 of a computer system 200 comprising a processor 202 and memory 203. The SMS comprises or is operatively coupled via a network connection to a plurality of storage resources R01-R09 and a plurality of resource managers RM1-RM5. The SMS 201 comprises a control module 213 and a scoring module 216 for calculating the score in step 106. The SMS 201 has access to a first mapping 212 and a second mapping 211 stored on storage medium 215. The SMS is adapted to access a plurality of requirements (SLOs) 210 assigned to one or more service classes, and to access a general mapping 209. The computer system 200 further comprises an interface 208, e.g. a network card, for receiving storage service requests from a client 205 via a network 207. According to other embodiments (not shown), the components of the computer system 200, in particular
the resources R01-R09 and resource managers RM1-RM5 may be hosted by other processing devices, thereby reducing the load of computer system 200.

**Figure 3** provides an overview of the plurality of mappings and their respective functions. A general mapping 209, which may be an editable configuration file or may be hard-coded within each of the resource managers, assigns all storage resources controllable by a particular resource manager to said resource manager. Thus, all storage resources comprising proprietary method interfaces 311 said resource manager is operable to call may be mapped in the general mapping to said resource manager. The first mapping 212 maps each service class to one or more resource managers. A parameter value within the first mapping of a given service class thereby indicates if all storage resources assigned to a given resource manager according to the general mapping 209 shall per default be assigned to the service class, i.e., shall be used for calculating a score and possibly provide storage capacity according to the requirements of said given service class. A second mapping 211 assigns or unassigns individual storage resources directly to or from one or more service classes. A second parameter value within the second mapping indicates whether the assignment between a given service class and a given storage resource in the second mapping should have the effect of using said storage resource for providing storage capacity in accordance with the requirements of said given service class or if said assignment should have the effect of excluding that said resource from a per-default assignment specified in the first mapping.

In the following, an embodiment will be described in greater detail in respect to the underlying mappings. A SMS according to said embodiment may comprise or be operatively coupled to a plurality of storage resources respectively being of a particular storage type, having been provided by a particular vendor and comprising at least one main functionality. For each unique combination of said property values, a corresponding resource manager is created and added to the SMS for controlling the respective storage resource:
<table>
<thead>
<tr>
<th>Resource Manager</th>
<th>vendor</th>
<th>model</th>
<th>Functionality</th>
<th>Storage Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SnapshotFileForIBMSONAS</td>
<td>IBM</td>
<td>SONAS</td>
<td>Snapshot</td>
<td>File</td>
</tr>
<tr>
<td>AsyncMirrorFileForIBMSONAS</td>
<td>IBM</td>
<td>SONAS</td>
<td>Asynchronous Mirroring</td>
<td>File</td>
</tr>
<tr>
<td>AsyncMirrorSnapshotFile-</td>
<td>IBM</td>
<td>SONAS</td>
<td>Snapshot with Asynchronous Mirroring</td>
<td>File</td>
</tr>
<tr>
<td>HighPerformanceFileForIBMSONAS</td>
<td>IBM</td>
<td>SONAS</td>
<td>Solid State Disks</td>
<td>File</td>
</tr>
<tr>
<td>SnapshotFileForNetApp</td>
<td>NetApp</td>
<td>FAS2000</td>
<td>Snapshot</td>
<td>File</td>
</tr>
<tr>
<td>FlashCopyBlockForIBMDS8000</td>
<td>IBM</td>
<td>DS8000</td>
<td>Snapshot</td>
<td>Block</td>
</tr>
<tr>
<td>GlobalMirrorBlockForIBMDS8000</td>
<td>IBM</td>
<td>DS8000</td>
<td>Asynchronous Mirroring</td>
<td>Block</td>
</tr>
<tr>
<td>HighPerformanceBlockForDS8000</td>
<td>IBM</td>
<td>DS8000</td>
<td>Solid State Disks</td>
<td>Block</td>
</tr>
<tr>
<td>FlashCopyBlockForIBM4SVC</td>
<td>IBM</td>
<td>SVC</td>
<td>Snapshot</td>
<td>Block</td>
</tr>
<tr>
<td>GlobalMirrorBlockForIBM4SVC</td>
<td>IBM</td>
<td>SVC</td>
<td>Asynchronous Mirroring</td>
<td>Block</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 1: distinct resource managers for distinct property value sets

The resource managers comprise program logic for controlling, i.e., managing and/or configuring, the functionalities of the supported storage resources in accordance with said resource’s storage type and device characteristics. For example, resource managers supporting storage type "File" implement generic interface methods for creating file systems, creating shares so remote clients can access the file system, deleting shares and file systems, etc. Likewise, resource managers supporting storage type "Block" implement generic interface methods for the configuration of LUN creation, mapping and masking depending on the requested access details for the block device, etc. The resource manager-specific implementation of these methods takes care of performing any additional, vendor and model-specific configuration of the respective storage resource in order to cause the created storage capacity unit (for instance file system or LUN) to expose the functionalities advertised by the resource manager via correspond-
ing generic method interfaces and to provide said functionalities to the client having submitted the storage service request. The type of storage capacity units is not restricted to file systems or IUNs, but may also be any other type of storage capacity like object-based or other types of storage. The resource managers are mapped to service classes in a first mapping 212. For example, above resource managers may be mapped to service classes having been defined by a cloud service provider as follows:

<table>
<thead>
<tr>
<th>Service Class</th>
<th>Resource manager</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediumAvailabilityForFiles</td>
<td>SnapshotFileForIBMxSONAS</td>
<td>yes</td>
</tr>
<tr>
<td>MediumAvailabilityForFiles</td>
<td>SnapshotFileForNetApp</td>
<td>yes</td>
</tr>
<tr>
<td>HighAvailabilityForFiles</td>
<td>AsyncMirrorFileForIBMxSONAS</td>
<td>yes</td>
</tr>
<tr>
<td>HighestAvailabilityForFiles</td>
<td>AsyncMirrorSnapshotFileForIBMxSONAS</td>
<td>yes</td>
</tr>
<tr>
<td>HighPerformanceForFiles</td>
<td>HighPerformanceFileForIBMxSONAS</td>
<td>yes</td>
</tr>
<tr>
<td>HighPerformanceForFiles</td>
<td>HighPerformanceFileForNetApp</td>
<td>yes</td>
</tr>
<tr>
<td>FlashCopyBlock</td>
<td>FlashCopyBlockForIBMDS8000</td>
<td>yes</td>
</tr>
<tr>
<td>FlashCopyBlock</td>
<td>FlashCopyBlockForIBMxSVC</td>
<td>yes</td>
</tr>
<tr>
<td>GlobalMirrorBlock</td>
<td>GlobalMirrorBlockForIBMDS8000</td>
<td>yes</td>
</tr>
<tr>
<td>GlobalMirrorBlock</td>
<td>GlobalMirrorBlockForIBMxSVC</td>
<td>yes</td>
</tr>
<tr>
<td>HighPerformanceBlock</td>
<td>HighPerformanceBlockForDS8000</td>
<td>no</td>
</tr>
</tbody>
</table>

Table 2: first Mapping of service classes and resource managers

For each of the service classes the service provider may provide a high-level, product-agnostic description of the kind and level of service it represents. The list of service classes along with these descriptions may then be exposed to service consumers which may choose a service class from this list and reference it when requesting storage capacity in accordance with a certain class of service.

The ‘default’ column represents the first parameter value being indicative if all storage resources controllable by a given storage manager should be selected for retrieving monitoring data for determining the storage capacity available for a requested service class. Depending on the configuration of the storage resources, each of the resource managers may support all
storage devices of the vendor's environment which they have been created for by default (if the first parameter value equals ‘yes’ implying ‘All assigned per default’). If the first parameter value has assigned the data value ‘no’ implying ‘NONE assigned per default’, none of the storage resources supported by a given storage manager is per default (but possibly by means of the second mapping) selected for calculating a score value and/or determining the storage capacity available for a given service request. The assignment of service classes to storage resources may further be specified in a second mapping 211 as exemplary depicted in table 3, wherein individual storage resources are respectively assigned to or are unassigned from one or more service classes:

<table>
<thead>
<tr>
<th>Service Class</th>
<th>Storage Resource</th>
<th>Assigned / Unassigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>IBM-SONAS-private-12345</td>
<td>Unassigned</td>
</tr>
<tr>
<td>all</td>
<td>NetApp-FAS2000-1382</td>
<td>Unassigned</td>
</tr>
<tr>
<td>HighPerformanceForFiles</td>
<td>NetApp-FAS2000-1382</td>
<td>Assigned</td>
</tr>
<tr>
<td>HighPerformanceBlock</td>
<td>StoragePool-HiPerf-2735</td>
<td>Assigned</td>
</tr>
</tbody>
</table>

Table 3: second mapping of service classes to storage resources

The "Assigned/Unassigned" column comprises a list of values of a second parameter being indicative of an assignment or un-assignment of a given storage resource to one or more (or even ‘all’) service classes.

All storage resources controllable by a resource manager mapped according to the first mapping depicted by table 1 to a requested service class are selected for calculating the score and/or for determining the storage capacity available for a given service class in case the first parameter value is indicative of a per-default mapping. For example, the resource manager SnapshotFileForIBMSONAS may have assigned according to a general mapping 209 a plurality of storage resources controllable by said resource managers, among them storage resource "IBM-SONAS-private-12345". According to the first mapping (table 2), resource manager SnapshotFileForIBMSONAS is assigned to service class MediumAvailabilityForFiles, whereby said mapping is flagged by a data value default='yes’. Thus, per default, storage capaci-
ity provided by any one of the storage resources mapped to \texttt{MediumAvailabilityForFiles} may be used for score calculation and capacity determination in case of a service request for class \texttt{MediumAvailabilityForFiles} was received. However, storage resource “IBM-SONAS-private-12345” will not be selected for said tasks as it has been unassigned from all service classes in the second mapping. No table corresponding to the general mapping is shown. The general mapping may be based on a dynamic execution of an evaluation rule etc. For example, the general mapping may comprise a program routine determining all resource managers of table 1 whose name ends with ‘SONAS’ and consider each SONAS storage resource as a storage resource being controllable by said resource manager.

Likewise, by default, all NetApp FAS2000 storage resources will be used in response to requested service classes mapped to any of the resource managers listed in table 1 whose name ends with “NetApp”. However, the specific storage resource NetApp-FAS2000-1382 will not be selected for score calculation or storage capacity allocation as said storage resource is unassigned in the second mapping from “all” service classes except for the service class \texttt{HighPerformanceForFiles}. Due to the explicit assignment in the second mapping (table 3), storage resource NetApp-FAS2000-1382 will only be selected in case a storage request for a \texttt{HighPerformanceForFiles} service class was received. Service class \texttt{HighPerformanceBlock} is not marked as a "default" service class in table 3 above, but is assigned explicitly to one specific storage resource, a particular storage pool. So requests for storage capacity which ask for service class "HighPerformanceBlock" will be handled by allocating LUNs from pool "StoragePool-HiPerf-2735", as specified in table 3, provided the capacity of this pool is sufficient.

\textbf{Figure 4} is a block diagram showing the resource managers RML-RM5” and storage resources R01-R09 in greater detail. Service classes 301-302 are mapped to resource manager RML in the first mapping. Service class I may comprise a requirement that data stored to a storage capacity provided in accordance with this service class should be stored permanently. Service class II may comprise a requirement that data older than a predefined time
interval should be considered as expired and be deleted by any storage service providing storage capacity in accordance with this service class.

Data expiration: Storage resource R01 differs from R02 in that its configuration A 315.1 causes a deactivation of the data-expiration functionality F_I (indicated by a dotted border) while configuration B 315.2 activates said functionality F_I (indicated by thick border line). By means of the first and second mapping, storage capacity of storage resource R01 but not of storage resource R02 is provided in response to a request for service class 301 while storage capacity of storage resource R02 but not of storage resource R01 is provided in response to a request for service class 302.

Calling storage type specific functions: Storage resources R01 and R02 are controllable by resource manager RM01 as both share the same model and storage type and were derived from the same vendor. Accordingly, their respective proprietary interfaces 311.1 and 311.2 are identical and can both be accessed by resource manager RM1. Each resource manager can be called by the SMS via its generic API 306 comprising method interfaces which have the same method signature if they relate to the same underlying storage function. For example, the signature of method interfaces 308.1 and 308.2 for “providing file system based data access” may be identical although the underlying functionality 314.2, 321 is provided by proprietary program routines 312.2, 323. Storage resources R01(R02) and R03 were obtained from different vendors, have different models, but share the same storage type. R01(R02) and R03 are adapted to provide file system based data access functionality F_II. Each resource manager comprises program routines 309, 318 being adapted to call a proprietary interface 311, 320 of a storage resource. Said routines 309, 318 are typically not generic but need to be adapted for each kind of supported storage resource. Thus, the method interfaces 308.1 and 308.2 for providing file system based data access are callable via a method interface 308.1, 308.2 having the same method signature, whereby calling function F_II via said generic method interfaces 308.1 and 308.2 results in the execution of proprietary program routines 314.2 and 321 called via proprietary interface methods PFI_II.1, PFI_II.2.
CLAIMS

1. A computer-implemented method for providing storage resources (R01-R09) of a storage management system (201), the method comprising:
   - receiving (101) a storage service request comprising an indication of a service class (301-303), a service class being a data object having assigned one or more requirements (210) to be fulfilled by any storage service provided in accordance with said service class;
   - determining (102) all resource managers (RM1-RM5) mapped to the indicated service class according to a first mapping (212), each resource manager being a runtime data object comprising a generic API (306, 307) providing control over one or more of the storage resources;
   - for each of the determined resource managers, determine (104) selected ones of all the storage resources controllable by said resource manager;
   - gathering (105) monitoring data of the selected storage resources, the monitoring data being indicative of free capacities of the selected storage resources;
   - comparing (106) the gathered monitoring data with the requirements of the indicated service class for calculating a score for each one of the selected storage resources; and
   - calling (107) at least one function of the resource manager operable to control the storage resource having an optimum score via said resource manager’s generic API, thereby configuring or initializing said storage resource to provide the requested storage service in accordance with the indicated service class.

2. The computer-implemented method of any one of the previous claims, further comprising:
   - upon adding a new storage resource to the storage management system, the new storage resource
     a) being of a new vendor of the storage resource, and/or
     b) being of a new storage type, and/or
c) being of a new storage model, and/or
d) offering a new function,

adding a new resource manager to the storage management system, the new resource manager comprising new executable routines, said routines:

a) being vendor-specific, and/or
b) being storage type specific, and/or
c) being storage model specific, and/or
d) implementing the new function,

thereby enabling the resource management system to configure or initialize said new storage resource for providing the new storage resource upon request.

3. The computer-implemented method of claim 2, further comprising:
   - for each unique combination of two, three or four out of a set of properties of the storage resources, providing a resource manager implementing executable routines particular to said unique combination of properties, said properties comprising:
     - the vendor, and/or
     - the storage type, and/or
     - the storage model, and/or
     - a function of said storage resource.

4. The computer-implemented method of any one of the previous claims, wherein the generic APIs of all storage managers operable to control storage resources sharing a function (F_II) share a method interface (MI_II), said method interface enabling the storage management system (201) to control to said shared function.

5. The computer-implemented method of any one of the previous claims, wherein the generic APIs of all storage managers having assigned stor-
age resources sharing a storage type share one or more storage-type-specific method interfaces, said storage-type-specific method interfaces enabling the storage management system to control one or more respective storage-type-specific functions.

6. The computer-implemented method of any one of the previous claims, wherein the determination of the selected ones of all the storage resources controllable by any one of the determined resource managers comprises:
   - accessing the first mapping, wherein each pair of mapped service class and resource manager is flagged by a first parameter value;
   - in case said first parameter value of any one of said pairs has a first data value (yes), using all storage resources controllable by the resource manager of said pair as the selected storage resources of said resource manager;
   - in case said first parameter value of the given pair has a second data value (no), not using any of the storage resources controllable by the resource manager of said pair as the selected storage resource of said resource manager.

7. The computer-implemented method of any one of the previous claims, wherein the determination of the selected ones of all the storage resources controllable by one of the determined resource managers comprises:
   - accessing a second mapping (211), the second mapping comprising pairs of service classes and storage resources mapped to each other, wherein each of said pairs is flagged by a second parameter value;
   - in case said second parameter value of any one of said pairs has a third data value (ASSIGNED), using the storage resource of said pair as one of the selected storage resources;
   - in case said second parameter value of said pair has a fourth data value (UNASSIGNED), not using the storage resource of said pair as the selected storage resources.
8. The computer-implemented method of claim 7, wherein the selected ones of the storage resources are determined in accordance with the first and the second mapping, wherein the mapping information in the second mapping for determining the selected storage resources overrule the respective mapping information in the first mapping.

9. The computer-implemented method of any one of the previous claims, comprising:

- generating a list of service classes supported by the storage management system;

- providing interface means for outputting the list of service classes and for allowing a requestor of the storage service to select one of the service classes from the list and to include an indication of the selected service class into the storage service request submitted via said interface.

10. The computer-implemented method of any one of the previous claims, comprising:

- adding a new service class associated with new requirements;

- in case one or more of the resource managers are already operable to control a storage resource for providing said storage resource in accordance with the new requirements, mapping the new service class to said one or more resource managers in the first mapping;

- in case none of the resource managers is operable to control a storage resource for providing said storage resource in accordance with the new requirements, creating a new resource manager operable to control at least one of the storage resources for providing said at least one storage resource in accordance with said new requirements and mapping the new resource manager to the new service class in the first mapping.
11. The computer-implemented method of claim 10, the adding of the new service class being executed upon adding a new storage function to the storage management system, wherein the new requirements are requirements in respect to said new storage function.

12. The computer-implemented method of any one of the previous claims, comprising:
   - for each one of all service classes supported by the storage management system:
     ▪ selecting a current one of the service classes;
     ▪ determining the resource managers assigned to the current service class according to the first mapping;
     ▪ determining selected ones of all the storage resources controllable by said determined resource managers;
     ▪ gathering monitoring data of the selected storage resources for providing monitoring data of the current service-class;
   - assembling the monitoring data of all service classes for outputting a report.

13. The computer-implemented method of any one of the previous claims, wherein the gathering of monitoring data of the selected storage resources is executed by one of the resource managers being operable to control said selected storage resources and being mapped according to the first mapping to the service class indicated in the service request or being the current service class.

14. A computer-readable medium (215) comprising computer-readable program code embodied therewith which, when executed by a processor, cause the processor to execute a method according to anyone of the previous claims.

15. A computer system (200) operatively coupled to a plurality of storage resources (R01-R09), the computer system comprising:
- an interface (208) for receiving a storage service request;
- a storage management system (201) for managing the storage resources, the storage management system being adapted for:
  - receiving (101) the storage service request via the interface, the storage service request comprising an indication of a service class, a service class (301-303) being a data object having assigned one or more requirements (SLOs) to be fulfilled by any storage service provided in accordance with said service class;
  - determining (102) all resource managers mapped to the indicated service class according to a first mapping (212), each resource manager being a runtime data object comprising a generic API (306, 307) providing control over one or more of the storage resources;
  - for each of the determined resource managers (RM1-RM5), determine (104) selected ones of all the storage resources controllable by said resource manager;
  - gathering (105) monitoring data of all selected storage resources, the monitoring data being indicative of free capacities of the selected storage resources;
  - comparing (106) the gathered monitoring data with the requirements of the indicated service class for calculating a score for each one of the selected storage resources; and
  - calling (107) at least one function of the resource manager operable to control the storage resource having an optimum score via said resource manager’s generic API, thereby configuring or initializing said storage resource to provide the requested storage service in accordance with the indicated service class.
Application No: GB1209209.4
Examiner: Jake Collins
Claims searched: 1-15
Date of search: 13 September 2012

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

<table>
<thead>
<tr>
<th>Category</th>
<th>Relevant to claims</th>
<th>Identity of document and passage or figure of particular relevance</th>
</tr>
</thead>
</table>
| X        | 1-15               | US 7945640 B1  
(VANTINE) See whole document in particular column 8 lines 40-67) |
| X        | 1-15               | US 2007/0283119 A1  
(SARKAR ET AL) See paragraphs 35 and 87-89 in particular |
| A        | -                  | Dynamic Storage Resource Management Framework for the Grid; MIN  
22nd IEEE / 13th NASA Goddard Conference on Monterey, CA, USA  
11-14 April 2005  
available on-line at:  
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1410748 |

Categories:

| X | Document indicating lack of novelty or inventive step |
| Y | Document indicating lack of inventive step if combined with one or more other documents of same category. |
| & | Member of the same patent family |
| A | Document indicating technological background and/or state of the art. |
| P | Document published on or after the declared priority date but before the filing date of this invention. |
| E | Patent document published on or after, but with priority date earlier than, the filing date of this application. |

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC:X :

Worldwide search of patent documents classified in the following areas of the IPC
G06F; H04L

The following online and other databases have been used in the preparation of this search report

WPI, EPODOC, XPESP, INSPEC, XPAIP, XPI3E, XPIEE, XPIOP, XPESP2

International Classification:

<table>
<thead>
<tr>
<th>Subclass</th>
<th>Subgroup</th>
<th>Valid From</th>
</tr>
</thead>
<tbody>
<tr>
<td>G06F</td>
<td>0003/06</td>
<td>01/01/2006</td>
</tr>
<tr>
<td>G06F</td>
<td>0017/30</td>
<td>01/01/2006</td>
</tr>
</tbody>
</table>