Efficient and agile storage management in software defined environments

A. Alba
G. Alatorre
C. Bolik
A. Corrao
T. Clark
S. Gopisetty
R. Haas
R. I. Kat
B. S. Langston
N. S. Mandagere
D. Noll
S. Padbidri
R. Routray
Y. Song
C.-H. Tan
A. Traeger

The IT industry is experiencing a disruptive trend for which the entire data center infrastructure is becoming software defined and programmable. IT resources are provisioned and optimized continuously according to a declarative and expressive specification of the workload requirements. The software defined environments facilitate agile IT deployment and responsive data center configurations that enable rapid creation and optimization of value-added services for clients. However, this fundamental shift introduces new challenges to existing data center management solutions. In this paper, we focus on the storage aspect of the IT infrastructure and investigate its unique challenges as well as opportunities in the emerging software defined environments. Current state-of-the-art software defined storage (SDS) solutions are discussed, followed by our novel framework to advance the existing SDS solutions. In addition, we study the interactions among SDS, software defined compute (SDC), and software defined networking (SDN) to demonstrate the necessity of a holistic orchestration and to show that joint optimization can significantly improve the effectiveness and efficiency of the overall software defined environments.

Introduction

Software defined environment and software defined storage

Rapidly changing IT (information technology) environments are posing a challenge, in particular to the traditional processes of storage management and provisioning. Massive data growth is being compounded by less predictable and increasingly diverse needs in terms of usage of that data. In such conditions, the long software installation and hardware configuration cycles dependent on manual and expert operations invariably lead to poor resource utilization and lack of responsiveness from an application perspective. More generally, trends such as reduced application deployment time, increasing use of commodity components, and simplification of IT operations are all reflections of important business needs. In terms of storage, this means keeping investment and exploitation costs from rising overwhelmingly with the growth of data, while at the same time increasing agility to respond to changing business demands in a cost-effective manner.

The advent of software defined environments (SDEs) comes as a response to these needs. In essence, an SDE automatically allocates workloads to the most suitable set of infrastructure resources, dealing with potential heterogeneity across both workloads and resources, and continuously optimizes such allocations to account for changes in workload needs and resources.

Software defined storage (SDS) refers broadly to the evolution of storage systems to meet the needs of SDE, affecting the design of control and data planes. For instance, on the control plane, traditional storage systems can expose interfaces to automate the provisioning of their resources. On the data plane, workloads can optimize their data accesses through explicit interactions with the storage system, as discussed in [1]. Finally, data plane functions may be moved altogether into a software stack that can be instantiated on commodity hardware to achieve lower costs, as discussed elsewhere [2].

Digital Object Identifier: 10.1147/JRD.2014.2302381

0018-8646/14 © 2014 IBM

In the very dynamic context of SDEs, storage is however a notable exception: whereas failures of compute or network resources can be mitigated by repeating computations or resending data, stored data cannot be re-created in such a straightforward manner. Furthermore, data may have to remain persistent after the lifecycle of the workload. Storage resources must therefore offer the means to satisfy a certain level of data protection, in addition to, for instance, availability or performance levels also applicable to the other types of resources. Moreover, depending on its size or sensitivity, persistent data will also limit the mobility of workloads that otherwise encapsulate their needs in a form operable by various infrastructure providers.

Software defined storage objectives

An analogy with software defined networking (SDN) is helpful to reach a more formal definition of the objectives associated with SDS. SDN is aimed at providing a software implementation for some or all of the networking control plane in standard servers, and use of a programmatic interface to the data plane, which itself may even be implemented in software [3, 4]. The SDN approach can therefore be divided into two dimensions, both of which apply to SDS as well, as illustrated in the following examples: (1) a horizontal dimension of global optimization across nodes, where control plane functions shift from a fully distributed model to a centralized model in order to enable global optimization across the nodes; and (2) a vertical dimension of the level of software integration of the control and data planes. As an example of global optimization applied to SDS, consider storage systems where an individual storage system level tiering policy [5] is configured. A global tiering optimization policy across multiple storage systems would be able to rebalance the load by moving data to a more idle storage system, possibly passing along the original tiering information of that data to improve placement within the new system. As an example of increasing levels of software integration applied to SDS, consider the typical RAID (Redundant Array of Independent Disks) function: Initially, only a static configuration of Logical Unit Number (LUNs) with preconfigured RAID levels is presented by the hardware (storage system) to the software (running in the servers); then, software becomes capable of configuring LUNs with desired RAID levels through a programmatic interface to the hardware; finally, the RAID function is performed in software, eliminating the hardware availability restriction and enabling optimizations such as distributed RAID across nodes.

The following example illustrates the goals associated with SDS at a high level, contrasting the existing situation with the expected benefits of SDS. Currently, an application deployment starts by having the application designer state the expected storage capacity and performance requirements, followed by having the storage administrator define logical volumes for the required storage capacity and map them to the application. However, the application lifecycle is dynamic, and will require the storage administrator to manually adjust for various events: storage capacity needs to be increased or decreased; application performance degrades due to resource contention; performance requirements change (increase or decrease); data protection needs change; replication policies change; Recovery Point Objective (RPO) and Recovery Time Objective (RTO) of the data changes; backup and archive policy changes. In the future, applications are expected to specify storage requirements explicitly, including aspects related to performance, capacity, RPO/RTO, and replication. An orchestration component will automatically identify and configure the appropriate compute, network, and storage resources needed to satisfy the requirements. If the performance of an application is impacted, the orchestration component will automatically detect it and adjust resources to satisfy the requirements. If the requirements change, applications will indicate that to the orchestration component so resources are adjusted accordingly.

This example captures both the horizontal dimension of global optimization achieved across compute, network, and storage resources, as well as the vertical dimension of a programmatic API (application programming interface) to automatically configure and adapt resource requirements as changes are detected. It also shows on-demand configuration of storage infrastructure capabilities based on application/workload requirements. By abstracting infrastructure capabilities into generic storage capabilities independent of device- or vendor-specific capabilities, consumers of a storage service are decoupled from infrastructure details. Likewise, storage service providers gain flexibility as to which storage resources are used for fulfilling a given storage service request, characterized by capacity and required service class. Service providers automate the storage allocation process by defining a catalog of service classes, and specifying in configuration templates how storage within each class is allocated for the various types of storage resources available.

The key contribution of this paper is a novel SDS solution named IBM Open Platform for Storage, which embraces the open ecosystem based on OpenStack** with extensible and interoperable APIs. In addition, it enables seamless integration with enterprise storage management tools. Our solution provides a unified storage control plane that enables SDS functionalities and serves as an integral component of the software defined environment landscape.

The remainder of the paper is organized as follows. We first summarize the state-of-the-art SDS solutions and discuss their advantages and shortcomings. Next, our SDS solution, i.e., IBM Open Platform for Storage, is introduced, and detailed descriptions of SDS capabilities are provided. In addition, the orchestration among our SDS solution and other components in SDE is discussed. Finally, we use a case study of IBM Connections deployment in an SDE lab to illustrate the benefits of our SDS solution and conclude the paper.

State-of-the-art software defined storage solutions

This section discusses related work relevant to the subject of this paper. The emerging SDS solutions can be categorized into two groups based on their level of openness and the existence of associated ecosystems.

Software defined storage in enterprise solutions

For years, the storage industry has been facing challenges in managing heterogeneous storage devices which typically have their own management interfaces and protocols. Among the first solutions was the Storage Management Initiative Specification (SMI-S) [6] proposed by the Storage Networking Industry Association, which defined interoperable unified constructs to ease management tasks across heterogeneous storage devices. However, in order to achieve the objectives of SDS as presented in the previous section, additional enhancements are needed in the storage management layer to enable dynamic, agile, and highly automated solutions. In particular, the SMI-S defines interfaces at the data-plane layer, but does not address the SDS requirement of storage infrastructure capabilities to be represented in service-oriented, device-agnostic form, so that application requirements can be mapped to these capabilities.

As an example for an industry effort to close the gap between SMI-S and SDS, the IBM Virtual Storage Center (VSC) [7], with components of SAN (storage area network) volume controller (SVC) [8] for storage virtualization, Flash Copy Manager [9] for application-aware snapshot management, and the IBM storage resource management software product Tivoli Storage Productivity Center (TPC) [10], introduced a Storage Management API for Clouds (SMAC), which provides for a separation of concerns between storage service providers and storage service consumers. The SMAC API allows storage consumers to request new storage capacity by communicating the storage service requirements of this new capacity in abstract and service-oriented terms, without requiring them to have any knowledge about the resources in the storage infrastructure, such as device-specific capabilities. The entities surfaced by the SMAC API are (1) *Service Class*, which serves a dual purpose: for the provider, it represents storage management capabilities present in the storage infrastructure in abstract and device-independent terms; for the consumer, it captures the storage service requirements being requested for a given application; (2) *Capacity Pool*, which groups storage resources to satisfy new storage requests. These pools may be organized, for instance, along organization hierarchy, geography or by service classes the storage resources in the capacity pool support; and (3) *Delivery Unit*: a specific existing storage capacity, such as a storage volume or NAS share, along with its storage service requirements. A storage service consumer will request new delivery units, specifying the service class needed for the new delivery unit, and from which capacity pool it should be provisioned. TPC then determines the most appropriate storage resource it knows about from the specified capacity pool which fulfills the specified service class, in order to place the new delivery unit.

Further conceivable enhancements to TPC and the SMAC API include addressing the need for improved elasticity and flexibility. Also, a more comprehensive implementation of the various possible categories of storage service requirements, such as availability, security, and retention, is needed for a more complete establishment of an SDS control plane, which also serves as one of the motivations of our work as will be introduced shortly.

As of this writing, the specification of the SMAC API is not yet publically disclosed, but is used internally by other IBM offerings. An example offering making use of the SMAC API is the IBM SmartCloud* Storage Access [11] that implements a solution for building a private storage cloud, including abilities to monitor space consumption by tenant and enabling chargeback to them for the capacity they use. Other IBM products utilizing SMAC for automated, service-oriented infrastructure provisioning including storage are Tivoli Service Automation Manager [12] and IBM SmartCloud Orchestrator [13].

Other vendors have also been active in closing the gap between storage infrastructure management as enabled by SMI-S, and providing an SDS solution. Examples for these activities available in the market include EMC ViPR** [14], NetApp ONTAP** [15], Coraid EtherCloud Storage Manager** [16], among many others. In [17], an SDS architecture is proposed on a Windows**-based I/O stack to facilitate fine-grained I/O service differentiation. However, so far none of these provide the functional or device support coverage required for establishing a comprehensive SDS solution.

Software defined storage in the open source community

OpenStack is an open source cloud management project, which has seen significant adoption and vendor participation. OpenStack provides a platform for SDS and creates levels of abstraction between the storage and how it is consumed by applications. At this time, OpenStack includes two storage components: *Swift*, providing object storage to applications; and *Cinder*, providing block storage for VMs (virtual machines). Support for making file storage visible to VMs is on the roadmap of OpenStack, and key industry vendors, including IBM, are collaborating to provide this functionality.

OpenStack Swift is an object storage system, which manages and provides object storage to clients. The object storage is accessed via a REST (Representational state transfer) API, called Swift API, which is similar to Amazon S3 API or CDMI (Cloud Data Management Interface) [18]. Swift is architected to offer resilient and scalable storage, which automatically replicates data across available disks and nodes to provide scalability, availability and data protection. Swift is targeted to provide low cost storage as it can be run on storage rich commodity servers. Swift is also used by other OpenStack components. Swift can provide the VM image back-end repository for OpenStack image management and serve as a target for volume backups. Object storage is a very popular platform for new applications, especially those created and for the most part using resources provided by the Web due to the prevalent use of REST over HTTP. It provides the scalability and agility that file- and block-based storage is lacking because of the strict inherent legacy semantics embedded into them. Moreover, IBM Research has demonstrated advanced capabilities such as computational storage [19], metadata and search [20], and secure multi-tenancy for object storage [21].

OpenStack Cinder is the persistent storage management component in OpenStack. Cinder provides block-storage management, such as volume allocation and volume attachment to servers. A key feature of Cinder is a unified Cinder API to manage heterogeneous storage, and a set of pluggable backend drivers that interface with the storage systems. Today, Cinder manages high-end storage systems, such as IBM XIV* and SAN Volume Controllers, mid-range systems, like IBM Storwize* systems, distributed storage systems such as IBM GPFS* (General Parallel File System) [22] and even commodity storage using Linux** LVM (logical volume manager). Cinder includes a mechanism to report the capabilities and state of the storage system. Capabilities can include items such as media type, protocol, and compression support, while state can include available free space. Further, administrators can define “volume types” which define a named set of requirements. Users may then choose a volume type when creating a volume, providing requirements for allocating their storage. Cinder has a rudimentary “filter-scheduler” that compares the requirements supplied by the volume types with the capabilities provided by the storage system, and chooses a storage system that satisfies the requirements. OpenStack’s architecture allows for schedulers to be plugged in replacing the default scheduler, allowing 3rd party analytics tools to be utilized. OpenStack’s support for storage system capabilities and the ability to tag volumes by volume types is the first level of storage resource abstraction. This abstraction can be exploited by advanced management tools to provide SDS management. In the Havana release of OpenStack, Cinder supports volume migration between storage systems through volume migration features available in the storage system, or by leveraging volume migration support in KVM (Kernel-based Virtual Machine) [23].

OpenStack is an emerging platform with wide industry support, but is still lacking key features to address the complexity of enterprise applications, such as backup, disaster recovery, fabric management, advanced volume placement, and continuous storage optimization—challenges for which solutions are described in [24, 25]. Of the features listed above, initial solutions for backup and fabric management are emerging. IBM Tivoli Storage Manager (TSM), the IBM data protection product for open systems, has limited support for OpenStack starting in the Havana release. Fabric management is evolving with the introduction of availability zones in Cinder but still includes the assumption of any-to-any connectivity. In summary, OpenStack provides a viable platform for SDS but is still only an infrastructure-as-a-service solution as of today, which should be complemented with advanced solutions for workload management, service level management and information lifecycle management. In the next section, we will introduce the IBM Open Platform for Storage framework, which enjoys the flexibility and interoperability of the OpenStack ecosystem while providing enriched enterprise level storage management solutions by incorporating the salient features of the aforementioned VSC.

IBM Open Platform for software defined storage

The IBM Open Platform for Storage framework is one of the essential components in the IBM SDE architecture to provide SDS management functions. **Figure 1** illustrates a conceptual architecture of the overall SDE architecture. Various workloads are described using declarative workload abstraction methods to capture the application-specific requirements such as infrastructure architecture and business operational workflow. Another core component of the SDE is the resource abstraction layer which provides a unified interface for the provisioning, management, and monitoring of underlying resources in compute, networking, and storage domains. Therefore, the complexity of heterogeneous device management is transparent to users. The core SDE Unified Control Plane interprets the workload abstraction and manages the underlying resources to provide integrated services customized to the specific application requirements. In other words, it interprets the workload requirements, the abstraction of underlying resources, and orchestrates the SDC, SDN, and SDS components for agile and efficient management. Our proposed IBM Open Platform for Storage is the framework in SDE which enables SDS capabilities and interacts with counterparts of software defined compute and SDN. The framework provides a unified programmable control plane that allows Service-Level Objective (SLO)-oriented storage management on top of heterogeneous data planes. The framework is extensible and aligned with the OpenStack APIs for interoperability and consumption,

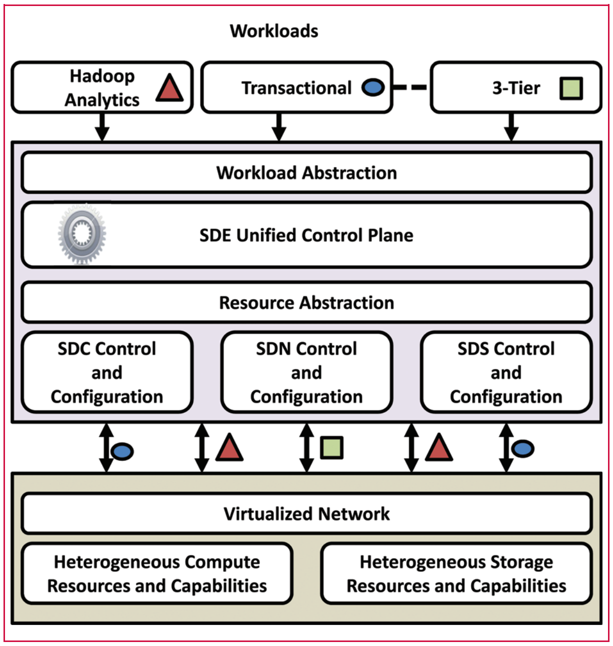

Workloads
Hadoop Analytics
Transactional
3-Tier
Workload Abstraction
SDE Unified Control Plane
Resource Abstraction
SDC Control and Configuration
SDN Control and Configuration
SDS Control and Configuration
Virtualized Network
Heterogeneous Compute Resources and Capabilities
Heterogeneous Storage Resources and Capabilities

Figure 1

Architectural view of software defined environments. The ovals, triangles, and squares represent different types of workloads.

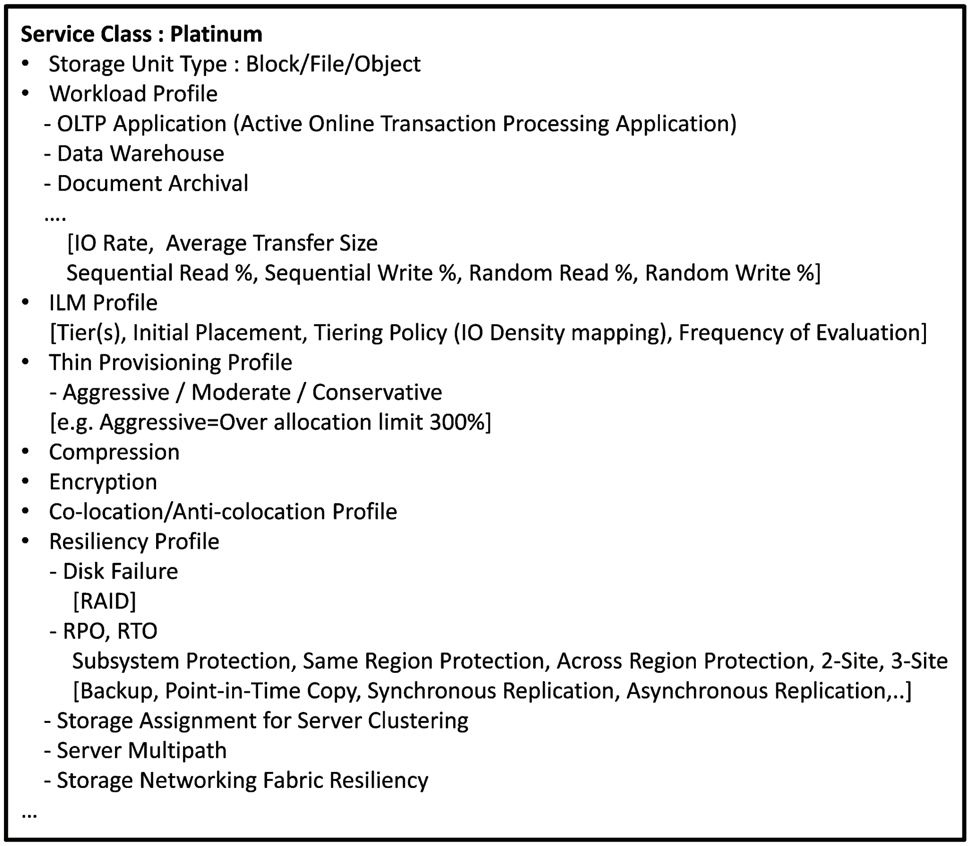
which is also the objective of the overall SDE architecture. In addition, our solution provides a pluggable interface to integrate with enterprise storage management products by implementing a Cinder filter scheduler. From a functional perspective, the framework building blocks include (1) workload abstraction, (2) resource abstraction, (3) workloads-resource mapping, and (4) continuous optimization. Next, we will provide detailed descriptions on each of the building blocks.

Workload abstraction

The rigid boundaries of compute, storage and networking in traditional IT environments impose significant challenges for efficient and agile application development and service deployment. In order to improve the portability and promptness of novel software and services delivery, declarative and expressive methods to capture the software and infrastructure requirements of the workload [26] are strongly desired where software patterns and infrastructure patterns can be tightly linked. In the open source community, TOSCA (Topology and Orchestration Specification for Cloud Applications) proposed by OASIS (Organization for the Advancement of Structured Information Standards) [27] and Linked Data format by OSLC (Open Service for Lifecycle Collaboration) [28] are popular methods to describe and manage the resources that applications require throughout their lifecycles. For example, a traditional three-tier web application may request a web server, a database server, an application server, and this software pattern will be captured in one of the declarative language formats, e.g., JSON (JavaScript** Object Notation) or XML (Extensible Markup Language). Orchestration engines such as IBM SmartCloud Orchestrator and OpenStack Heat will parse the specifications and orchestrate the deployment of resources on the underlying infrastructure accordingly. From a storage perspective, the framework creates extensions on both the workload descriptive tools and orchestration engines to expose sufficient and flexible storage semantics and primitives. For example, in the workload description, the application can specify the size of a storage volume, the desired service class, and its associated policy including, but not limited to, workload profile [e.g., I/O requirements of online transaction processing (OLTP) or batch processing], resiliency profile (e.g., Recovery Point Objective and Recovery Time Objective), among many others. Therefore, the framework allows the application developers and system administrators to explicitly specify their desired properties and features for underlying storage resources. Such user-specific storage requirements can also be adjusted throughout the application lifecycle as the workload changes, which significantly enhance agility and flexibility. **Figure 2** shows an example of the workload abstraction and its association with the resource abstractions. As shown in the example, the user can specify the type of storage requested, the workload profile (e.g., batch processing job or transactional workload), and desired storage features such as compression, encryption, tiering policies, among others.

Resource abstraction

As one of the key differentiating features of SDE, resource abstraction of heterogeneous resources is of paramount importance to enable unified infrastructure management and optimization. The IBM Open Platform for Storage supports a variety of underlying storage resources ranging from enterprise disk subsystems to commodity storage devices such as the GPFS GSS solution based on low cost storage disks. In addition, the framework is able to manage and aggregate disparate storage resources from Storage Area Network, NAS (Network-Attached Storage), or DAS (Direct Attached Storage) architectures and provides a unified storage resource abstraction layer which allows users to allocate storage in consumable units of a block, file, or object type. For example, the framework can create a file in the GPFS platform using low cost storage devices, expose the file as a block storage volume for virtual machine consumption, and use GPFS functions such as snapshot and file migration to manage the virtualized block device. The underlying management complexity is transparent to the end user and the storage resources utilization efficiency can be improved. Therefore, the framework provides a logical storage resource abstraction view which significantly reduces



Service Class : Platinum

- Storage Unit Type : Block/File/Object
- Workload Profile
  - OLTP Application (Active Online Transaction Processing Application)
  - Data Warehouse
  - Document Archival

  ….

  [IO Rate, Average Transfer Size
  Sequential Read %, Sequential Write %, Random Read %, Random Write %]
- ILM Profile
  [Tier(s), Initial Placement, Tiering Policy (IO Density mapping), Frequency of Evaluation]
- Thin Provisioning Profile
  - Aggressive / Moderate / Conservative

  [e.g. Aggressive=Over allocation limit 300%]
- Compression
- Encryption
- Co-location/Anti-colocation Profile
- Resiliency Profile
  - Disk Failure
    [RAID]
  - RPO, RTO
    Subsystem Protection, Same Region Protection, Across Region Protection, 2-Site, 3-Site
    [Backup, Point-in-Time Copy, Synchronous Replication, Asynchronous Replication,..]
  - Storage Assignment for Server Clustering
  - Server Multipath
  - Storage Networking Fabric Resiliency

…

Figure 2

Example of a service class and storage resource mapping.

the storage management complexity, achieves better storage resource utilization, and improves operational efficiency.

From a storage functionality perspective, the framework exposes two broad abstract categories of capabilities, i.e., (1) security and availability and (2) performance and optimization. In security and availability, storage capabilities such as authentication, auditing, encryption, data protection, high availability, among others are captured. The functions in performance and optimization include data striping, clustering, compression, de-duplication, and information lifecycle management (ILM, or tiering), which will be introduced in detail shortly.

It is worth noting that the way the framework invokes aforementioned functionalities varies according to the underlying storage infrastructure. For example, high-end enterprise storage subsystems have firmware-level advanced storage capabilities such as compression and de-duplication. In such scenarios, the framework uses these features by providing device-specific configurations using standardized or proprietary protocols. If the desired functionality is not available at the underlying storage devices, the framework invokes a software implementation to achieve the same functions such as compression and software-based RAID for data protection e.g. GPFS de-clustered RAID. From the end user's perspective, the implementation details are hidden and the platform is able to make optimal decisions based on business objectives and heterogeneous storage infrastructure to strike a balance between quality of service and cost.

Mapping to resources

The SDE Unified Control Plane obtains configuration information of the underlying infrastructure via resource abstraction. When a new request is received in a workload description format, the control plane will orchestrate among compute, networking, and storage for holistic resource provisioning. For storage resources, the control plane will extract the storage requirements in the workload description and pass the information to SDS for programmable workload to resource mapping. The storage resource mapping usually consists of three components, which will be introduced next.

Performance-aware storage placement

One of the key features enabled by the IBM Open Platform for Storage is the performance-aware storage placement capability. For each storage resource creation request, a "service class" needs to be specified, which captures the specific requirements for storage provisioning, e.g., RAID level and resiliency profile. Each service class represents a set of preconfigured storage configurations. The user can create new service classes, within the bounds of service class permutations supported by the service provider, which better characterize their specific workload. Note that the service class can be specified by workload abstraction tools using declarative TOSCA and OSLC formats. For example, a "Platinum" service class captures stringent storage requirements such as OLTP-High workload which requires low I/O latency with extensive random I/O accesses, and thus the storage volume created under this service class will be preferably placed on solid-state drive (SSD) instead of hard disk drive (HDD). In addition, if multiple storage pools are available to place the new storage volume, e.g., multiple SSD pools are available in the managed environment; the framework performs advanced storage performance analytics to decide which SSD pool should host the new volume. For example, when a new storage volume creation request is received with a "platinum" service class for OLTP-High workloads, the framework first analyzes the resource utilization conditions of all candidate SSD storage pools using historical data, and performs what-if analysis to predict the performance impact in terms of I/O load and latency by invoking the IntelliMagic/DiskMagic [29] whitebox model which can model the disk performance under various I/O activities. A simple strategy would be placing the new storage volume on the SSD which yields minimum performance impact, e.g., minimum response time changes. Other complementary performance-aware storage placement strategies, e.g., load balancing or minimum number of occupied storage devices, can be incorporated. It is also worth noting that the placement objective can also be configured per service class. Therefore, the coupling of service class and performance aware storage provisioning greatly enhances the flexibility and efficiency of storage provisioning where application-specific storage resource requirements can be specified, satisfied, and optimized.

Storage fabric management

Once the storage unit is created, the IBM Open Platform for Storage will establish the connection between the server and the storage volume. The framework provides a unified control plane for various storage networking techniques such as Fibre Channel, iSCSI (Internet Small Computer System Interface), InfiniBand**, and FCoE (Fibre Channel over Ethernet). The framework provides best practice-based zone management and fabric analytics to identify the best storage fabric configurations for servers and storage backends. For example, by analyzing the port utilization information on storage devices, the framework will intelligently provide load balancing for devices with multiple I/O ports to prevent over-utilization. Due to the combinatorial nature of the optimal fabric configuration problem, the framework employs an iterative hill-climbing algorithm to search the solution space and prune inefficient configurations until an optimal storage fabric configuration is identified.

Resiliency

Resiliency is of great importance for data protection and business continuity. The IBM Open Platform for Storage provides resiliency in the form of fabric resiliency and storage device level data protection. For fabric resiliency, the application can choose to have independent I/O paths from the compute server to the storage backends with no single point of failure. For storage device level data protection, the framework can allow the application to choose a replication strategy among several options such as point-in-time snapshot, synchronous mirroring, and asynchronous mirroring. Furthermore, the framework can provide application aware data protection by defining consistency groups to ensure resiliency. From the overall SDE viewpoint, the framework is an integral component providing end-to-end resiliency, which can obtain the compute domain information, e.g., server cluster setup for failover purpose, and then configure multi-path resiliency to guarantee that both the primary server and standby secondary servers can have reliable storage access to mitigate the impact of failures.

Continuous optimization

One example of continuous storage optimization provided by the framework is a concept known as storage information lifecycle management (ILM). The motivation is that the value of data is time-varying in its lifecycle. For example, the value of an email will decrease rapidly over time and it is more cost-effective to place archived email on low cost storage devices due to infrequent access, while the most frequently accessed data is placed on the high-end storage devices with low latency. The objective of storage ILM is to place the right data on the right storage tier at the right time. The placement of data on different storage resources based on I/O activities is also known as storage tiering. Note that while the data tier is categorized based on the historical I/O activities, the tier of storage devices are usually labeled by a manual process according to their performance and/or cost. Our framework allows applications to control storage ILM based on their unique needs. For example, each service class can specify the upper and lower bound of storage tier that will be used, and the automatic tiering policy can be specified in terms of I/O thresholds. Therefore, the IBM Open Platform for Storage provides granular control functions to explore the tradeoff between performance and cost for each application or workload. We will provide more examples of continuous storage optimization when describing an end-to-end scenario of SDE leveraging our framework capabilities.

SDE integration

Traditional data center IT solutions view compute, storage, and networking domains as separate management entities where different administration and management solutions are provided. However, due to the increasing demand for rapid and agile application service deployment, this traditional management paradigm significantly hinders the speed and efficiency of new IT service evolution. From a data center user's perspective, the perceived system performance experienced by applications depends on the integration of compute, networking, and storage. A holistic view of the data center with intelligent orchestration schemes will break the silo barriers of compute, network and storage, and significantly increase the quality of service and quality of user experience. In our SDE framework, the IBM Open Platform for Storage enables SDS functionalities and provides an integrated interface for orchestration with other components in SDE such as software defined compute and SDN. Our framework defines simple, yet rich programmable APIs (e.g., RESTful OSLC-compliant APIs) that allow joint optimization among various components in the SDE. In the following sections, we briefly discuss a few examples of the orchestration of SDE that can utilize the flexible interfaces the framework provides.

Storage-aware VM placement

Many VM placement algorithms have been proposed in the literature to solve the initial VM placement problem. Popular methods include CPU and memory aware placement, networking congestion aware placement, etc. In a data center with SAN storage architecture, a VM creation is usually associated with a vDisk (virtual disk) creation where the virtual disk is used to store the data for this VM and needs to be placed on the storage devices in the SAN. For I/O intensive applications, the storage access latency accounts for a significant portion of the total latency the application experiences. Therefore, the initial placement decision needs to consider where to place the VM on compute servers (i.e., CPU and memory resources) as well as the vDisk placement on the SAN, in a joint fashion. If the VM placement solution makes myopic decisions without obtaining sufficient knowledge and information from the storage domain, it is likely that the VM is placed on a server which has slow storage access, possibly due to a congested HBA/fabric switch on its I/O path. Therefore, the VM placement, especially for I/O intensive workloads, needs to consider all components along the I/O path comprising compute, networking, and storage, in order to make informed and efficient decisions. The framework provides flexible storage placement interfaces for the SDE Unified Control Plane, or simply the orchestrator, to make informed joint VM and vDisk placement decisions.

Integrated application resiliency

For many critical applications, resiliency is of great importance for business continuity. In order to mitigate the risk of unpredictable failures, redundant resources are provided for failover and fallback purposes. For example, a server cluster can be set up for an application where several servers are in standby mode and will replace primary servers in case of failures. From a storage perspective, data replication and high availability fabric configurations are important methods to handle unexpected device failures. The framework exposes such functionalities to allow the orchestrator to configure an end-to-end resiliency for the application across all domains in the SDE. It is worth noting that such decisions cannot be made without a holistic view. For example, it is possible that all servers in the server cluster share a same I/O path on the storage fabric. Therefore, while the resiliency in compute domain is assured, an integrated resiliency is missing. Our framework provides rich semantics to allow the orchestrator to achieve this comprehensive goal.

Service differentiation in SDE

In SDE, the resources are usually allocated according to the specific requirements of each application as well as its customizable service class. For example, a line of business application may enjoy better service in general compared to regular maintenance workloads. In our framework, each application can specify a service class which reflects the storage resource differentiation. The service class can be chosen from a set of predefined templates, e.g., OLTP-high or gold, or created according to the specific storage requirements. The framework provides sufficient flexibility which can facilitate differentiated services in the storage domain and can be orchestrated by the SDE control plane to provide a service differentiation across the data center.

Software defined storage in an SDE Lab and case studies

The proliferation and increased volatility of workloads and applications coupled with heterogeneous infrastructure requirements prohibits the manual deployment and tuning of applications. Automated and agile deployment is essential. In addition, continuous and optimal management of these workloads based on business policies is evident. To validate the constructs specified in this paper, and to facilitate the benchmarking of research results, IBM established an SDE lab consisting of scalable, heterogeneous infrastructure that includes PureFlex* servers [30], xSeries* and pSeries* servers interconnected with a high-speed switch fabric. The SDE lab divides the physical servers into Reusable Development Units, or RDUs, where each RDU comprises one or more physical servers according to the requirements of a tenant. As each RDU is allocated, the IP address and other RDU-specific information is defined in an XML file, which drives the Extreme Cloud Administration Toolkit (xCAT) [31] process that installs Ubuntu as the base operating system (OS). After the base OS is installed, control is passed to a Chef server [32], which uses community and customized recipes to install OpenStack and configure the network using additional parameters from the XML file. After the Chef recipes are complete, the RDU is usable and accessible via the OpenStack interface. Heterogeneous storage resources with a mix of enterprise-class, commodity, and direct attached storage are set up to provide the data plane. The control plane is enabled with a combination of OpenStack platform, our IBM Open Platform for Storage, and IBM Tivoli storage products.

Our demonstrations consist of multiple types of applications, e.g., Hadoop**/MapReduce analytics workloads, transactional OLTP workloads, three-tier web applications. To illustrate the application of SDE operations, we describe the "IBM Connections" workload [33], which is a three-tier web application. IBM Connections is a social software platform used internally by two hundred thousand IBM employees for business information exchange. Periodic brainstorming and knowledge exchange sessions such as the IBM InnovationJam* and the IBM Global Technology Outlook (GTO) drive very volatile workload requirements. Our SDE testbed emulates the scale, capabilities and operations of this environment under such volatile workload variations. From the SDS perspective, we demonstrate two major features of our platform: performance-aware storage placement and continuous storage analytics and optimization.

For performance-aware storage placement, the Connections application requires expertise in deploying and managing heterogeneous storage backends. In contrast, our platform can provide a unified storage control plane for various storage devices which are presented via resource abstraction. In addition, the Connections application consists of multiple components such as database, file sharing, archival, and each component has a different storage requirement. For example, the database component demands fast storage response time with high level of resiliency, whereas archival services may require low cost storage for cost efficiency and the response time is not a priority. In our framework, the Connections application can specify a workload description comprising "Database: OLTP-Platinum," "File Sharing: Gold," and "Archival: Silver," where the storage resource type is appended with its desired service class. The framework provides differentiated services for different service classes. OLTP-Platinum is always placed on tier-one storage first with fabric resiliency and metro-mirroring enabled, whereas the silver class will be placed on low cost storage. Therefore, our framework provides great flexibility and facilitates performance aware storage placement, differentiated via service classes.

For continuous storage analytics and optimizations, the framework can intelligently monitor and analyze the performance of each storage volume in order to meet the requirements of the Connections application. As introduced earlier, the Connections application will have fluctuating access patterns due to events such as IBM InnovationJam or Hackathon, during which the storage access will increase dramatically. Therefore, it is strongly recommended to place the storage volumes that are being accessed frequently on high end storage tiers with low response time and move them back when the access diminishes. Traditional storage management solutions need to manually plan the storage placement in advance to accommodate the fluctuation storage access pattern, and move storage volumes among various storage tiers following a predetermined process. In contrast, our framework provides intelligent ILM functionality that can monitor the I/O access of each storage volume, and automatically up-tier or down-tier the volume in line with the time-varying storage access patterns. The agility and automation enabled by the framework significantly reduces the management complexity. It is also worth noting that the automatic tiering policy is part of the storage requirements specified by workload description and captured by the service class. For example, it is likely that the database volume is critical in terms of overall quality of experience and thus favors a tiering policy that is aggressive in terms of up-tiering, compared to archival storage volumes. The framework provides fine-tuned customizability captured by service classes.

The scenarios discussed above focus on agile and flexible management of the underlying infrastructure resources. Integration with other development and operations (DevOps) and platform-as-a-service (PaaS) solutions are under investigation.

Conclusions and future work

In this paper, we investigated the emerging trend of SDS, which is an essential component of an SDE. The paradigm shift to SDE brings new challenges to storage provisioning and management. We provided an overview of the challenges involved as well as existing SDS solutions. As the focus of this paper, we presented the IBM Open Platform for Storage framework and described the design architecture and its unique features. In addition, we demonstrated the necessity of interlocking among storage, compute, and networking in software defined environments, where cross-silo joint optimizations can significantly boost the efficiency and performance of the data center environment.

The scenarios discussed in this paper focus on agile and flexible management of the underlying storage infrastructure. In the future, we are interested in exploring two research directions. First, we are investigating approaches to integrate our framework with other DevOps solutions. Second, we aim to align our solution with platform-as-a-service (PaaS) and software-as-a-service (SaaS) from a storage perspective.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of OpenStack Foundation, EMC Corporation, NetApp Inc., Coraid Inc., Microsoft Corporation, Linus Torvalds, Sun Microsystems, InfiniBand Trade Association, or Apache Software Foundation in the United States, other countries, or both.

References

1. M. Harsha, J. McCullough, G. Porter, R. Kapoor, S. Savage, A. Snoeren, and A. Vahdat, “scc: Cluster storage provisioning informed by application characteristics and SLAs,” in *Proc 10th USENIX Conf. FAST*, Berkeley, CA, USA, 2012, p. 23.
2. IBM Corporation, Armonk, NY, USAIBM GPFS Storage Server. [Online]. Available: http://www-03.ibm.com/systems/x/hardware/largescale/gpfsstorage/
3. Is OpenFlow/SDN Good at Forwarding? [Online]. Available: http://networkheresy.com/2011/11/17/is-openflowsdn-good-at-forwarding/
4. Software-Defined Networking: The New Norm for Networks. [Online]. Available: https://www.opennetworking.org/sdn-resources/sdn-library/whitepapers
5. S. Seshadri, P. H. Muench, L. Chiu, I. Koltsidas, N. Ioannou, R. Haas, Y. Liu, M. Mei, and S. Blinick, “Software defined just-in-time caching in an enterprise storage system,” *IBM J. Res. & Dev.*, vol. 58, no. 2/3, Paper 7, 2014 (this issue).
6. IBM Corporation, Armonk, NY, USA, Storage Management Initiative-Specification. [Online]. Available: http://www.snia.org/forums/smi/tech_programs/smis_home
7. IBM Corporation, Armonk, NY, USA, IBM SmartCloud Virtual Storage Center. [Online]. Available: http://www-03.ibm.com/software/products/us/en/vsc/
8. IBM Corporation, Armonk, NY, USA, IBM System Storage SAN Volume Controller. [Online]. Available: http://www-03.ibm.com/systems/storage/software/virtualization/svc/
9. IBM Corporation, Armonk, NY, USA, Tivoli Storage FlashCopy Manager. [Online]. Available: http://www-01.ibm.com/software/tivoli/products/storage-flashcopy-mgr/
10. IBM Corporation, Armonk, NY, USA, IBM Tivoli Storage Productivity Center. [Online]. Available: http://www-03.ibm.com/systems/storage/software/center/
11. IBM Corporation, Armonk, NY, USA, IBM SmartCloud Storage Access. [Online]. Available: http://www-03.ibm.com/systems/storage/software/smartcloud-storage-access/index.html
12. Tivoli Service Automation Manager. [Online]. Available: http://www-01.ibm.com/software/tivoli/products/service-auto-mgr/
13. IBM Corporation, Armonk, NY, USA, IBM SmartCloud Orchestrator. [Online]. Available: http://www-03.ibm.com/software/products/us/en/smartcloud-orchestrator/
14. EMC, Hopkinton, MA, USA, ViPR. [Online]. Available: http://www.emc.com/data-center-management/vipr/index.htm
15. NetApp ONTAP. [Online]. Available: http://www.netapp.com/us/products/platform-os/data-ontap-edge
16. Coraid, Redwood City, CA, USA. [Online]. Available: http://www.coraid.com/
17. E. Thereska, H. Ballani, G. OShea, T. Karagiannis, A. Rowstron, T. Talpey, R. Black, and T. Zhu, “IOFlow: A software-defined storage architecture,” in *Proc. ACM SOSP*, 2013, pp. 182–196.
18. CDMI. [Online]. Available: http://www.snia.org/tech_activities/standards/curr_standards/cdmi
19. E. K. Kolodner, S. Tal, D. Kyriazis, D. Naor, M. Allalouf, L. Bonelli, P. Brand, A. Eckert, E. Elmroth, S. V. Gogouvitis, D. Harnik, F. Hernandez, M. C. Jaeger, E. B. Lakew, J. M. Lopez, M. Lorenz, A. Messina, A. Shulman-Peleg, R. Talyansky, A. Voulodimos, and Y. Wolfsthal, “A cloud environment for data-intensive storage services,” in *Proc. IEEE CloudCom*, Athens, Greece, 2011, pp. 357–366.
20. M. C. Jaeger, A. Messina, M. Lorenz, S. V. Gogouvitis, D. Kyriazis, E. K. Kolodner, X. Su, and E. Bahar, “Cloud-based content centric storage for large systems,” in *Proc. FedCSIS*, Wroclaw, Poland, 2012, pp. 987–994.
21. M. Factor, D. Hadas, A. Hamama, N. Har’El, H. Kolodner, A. Kurmus, E. Rom, A. Shulman-Peleg, and A. Sorniotti, “Secure logical isolation for multi-tenancy in cloud storage,” in *Proc. 29th IEEE MSST*, Long Beach, CA, 2013, pp. 1–5.
22. F. Schmuck and R. Haskin, “GPFS: A shared-disk file system for large computing clusters,” in *Proc. USENIX FAST*, 2002, p. 19.
23. QEMU Live Block Migration. [Online]. Available: http://wiki.qemu.org/Features/LiveBlockMigration
24. S. Gopisetty, S. Agarwala, E. Butler, D. Jadav, S. Jaquet, M. Korupolu, R. Routray, P. Sarkar, A. Singh, M. Sivan-Zimet, C. Tan, S. Uttamchandani, D. Merbach, S. Padbidri, A. Dieberger, E. Haber, E. Kandogan, C. Kieliszewski, D. Agrawal, M. Devarakonda, K. Lee, K. Magoutis, D. Verma, and N. Vogl, “Evolution of storage management: Transforming raw data into information,” *IBM J. Res. & Dev.*, vol. 52, no. 4/5, pp. 341–352, Sep. 2008.
25. S. Gopisetty, E. Butler, S. Jaquet, M. Korupolu, T. Nayak, R. Routray, M. Seaman, A. Singh, C. Tan, S. Uttamchandani, and A. Verma, “Automated planners for storage provisioning and disaster recovery,” *IBM J. Res. & Dev.*, vol. 52, no. 4/5, pp. 353–366, Sep. 2008.
26. M. H. Kalantar, F. Rosenberg, J. Doran, T. Eilam, M. D. Elder, F. Oliveira, E. C. Snible, and T. Roth, “Weaver: Language and runtime for software defined environments,” *IBM J. Res. & Dev.*, vol. 58, no. 2/3, Paper 10, 2014 (this issue).
27. OASIS Topology and Orchestration Specification for Cloud Applications. [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca
28. Open Services for Lifecycle Collaboration. [Online]. Available: http://open-services.net
29. IntelliMagic, Leiden, The Netherlands. [Online]. Available: http://www.intellimagic.net
30. IBM Corporation, Armonk, NY, USA, IBM PureFlex Systems. [Online]. Available: http://www-03.ibm.com/systems/pureflex/pureflex_overview.html
31. IBM Corporation, Armonk, NY, USA, xCAT. [Online]. Available: http://www-03.ibm.com/systems/software/xcat
32. Chef. [Online]. Available: http://www.opscode.com/chef/
33. IBM Corporation, Armonk, NY, USA, IBM Connections. [Online]. Available: http://www-03.ibm.com/software/products/us/en/conn

Received August 15, 2013; accepted for publication September 16, 2013

Alfredo Alba *IBM Research - Almaden, San Jose, CA 95120 USA (aalba@us.ibm.com).* Dr. Alba is currently a Senior Software Engineer in the Storage Services Research Department. He provided professional services and advice to the IBM Global Services Division and its customers since 1997, an organization that he joined as a Staff Member in 2002. In 2005, he joined the IBM Research Division working on web-scale analytics platforms. He is particularly interested in semantic data integration modeling and large-scale distributed systems and analytics. He received bachelor’s and master’s degrees in physics from the Mexico Autonomous National University in 1994 and 1996, respectively, and in 2002, he received a Ph.D. degree in physics from Stanford University.

Gabriel Alatorre *IBM Research - Almaden, San Jose, CA 95120 USA (galatorr@us.ibm.com).* Mr. Alatorre is an Advisory Software Engineer in the Storage Services Research Department. He received a

B.S. degree in computer science from the Massachusetts Institute of Technology in 2006, and an M.S. degree in computer science from Stanford University in 2009. After finishing his graduate studies, he joined IBM at the Almaden Research Center, where he has worked on optimized storage placement and tiering tools. In 2012, he received an Outstanding Technical Achievement Award for his work on the storage optimization services and Research Division Award for his work on Tiered Storage Optimization in Tivoli Storage Productivity Center (TPC) product. He is author or coauthor of 11 patents. Mr. Alatorre is a member of the Society of Hispanic Professional Engineers (SHPE) in Silicon Valley.

Christian Bolik *IBM Software Group, IBM Research & Development Lab, Hechtsheimer Str. 2, 55131 Mainz, Germany (bolik@de.ibm.com).* Mr. Bolik is a member of the development team for the IBM Tivoli Storage Productivity Center (TPC) product. He is an IBM Master Inventor and the lead architect for the introduction of software defined storage concepts in TPC. Having joined IBM in 1996, his professional interest revolves around making storage resources and capabilities more easily consumable by storage consumers, enabling them to consume storage in service-like fashion. Mr. Bolik received a master's degree (Dipl.-Ing.) in computer systems engineering from the University of Hanover.

Ann Corrao *IBM Global Technology Services, Research Triangle Park, NC 27709 USA (anncorr@us.ibm.com).* Ms. Corrao is an IBM Distinguished Engineer in the IBM Global Technology Services division focused on Storage Services primarily for the strategic outsourcing business. She has been with IBM for 13 years. She has received an Outstanding Technical Achievement Award for her work in eBusiness hosting. She has authored or coauthored five issued patents and has filed several other patent applications. Ms. Corrao is a member of the Association for Computing Machinery.

Thomas Clark *IBM Software Division, Tucson, AZ 85744 USA (tkc@us.ibm.com).* Mr. Clark is an IBM Distinguished Engineer and Chief Architect for IBM Storage Software. He spent more than 11 years at Informix Software developing parallel database systems, joining IBM in 2001. Since joining IBM, his work has focused on file systems, applications in storage, and storage management software. Mr. Clark received a bachelor's degree in computer science and mathematics from Mt. Vernon Nazarene University and a master's degree in computer science from Portland State University.

Sandeep Gopisetty *IBM Research - Almaden, San Jose, CA 95120 USA (sandeep@us.ibm.com).* Mr. Gopisetty is an IBM Distinguished Engineer and a Senior Manager and has over 21 years of IBM Research experience. Mr. Gopisetty leads Storage Research as part of the Services Innovation Lab in the Almaden Research Center to address the high costs of labor and capital in strategic outsourcing accounts through effective management services. His crucial contributions in the area of storage and systems management have helped IBM become a leader in storage management products and technologies. Mr. Gopisetty's technical vision, leadership, and contributions have transformed our storage management products from a collection of related products into the integrated product, Tivoli Storage Productivity Center (TPC) with various advanced analytics. He is the recipient of several patents and IBM awards including Outstanding Innovation Award and several Supplemental Outstanding Technical Achievement Awards for his contributions to the architecture as well as leadership in driving his vision into plan and through implementation with a team that spanned three divisions in Storage Management. He has received his eleventh IBM Invention Plateau and holds 23 patents. Mr. Gopisetty also served on the program committee for several conferences and is a member of the IBM Academy of Technology.

Robert Haas *IBM Corporate Headquarters, Armonk, NY 10504 USA (rha@zurich.ibm.com).* Dr. Haas is in charge of storage strategy in the IBM Corporate Strategy Department. Prior to this assignment, he led research in storage security, distributed storage, and tape- and flash-based systems at the Storage Technologies Department at the IBM Zurich Research Lab. He received a Ph.D. degree from the Swiss Federal Institute of Technology (ETH), Zurich, in 2003, and an M.B.A. degree from Warwick Business School, United Kingdom, in 2011. Dr. Haas has authored over 60 scientific publications, technical standards, and patents.

Ronen I. Kat *IBM Research Division, IBM Haifa Labs, Haifa University Campus, Mount Carmel, Haifa 31905, Israel (ronenkat@il.ibm.com).* Dr. Kat is a Research Staff Member at the Storage Systems Group. He leads the development of storage advanced features for virtualization and cloud environments, including development of storage features for OpenStack and data reduction. His research interests include compression, storage in the cloud, energy usage, storage systems, distributed algorithms, and communication networks with emphasis on autonomic systems and self-stabilization. Before joining IBM, he worked in data center IT, providing services for mainframe performance, communications, databases, and applications, and provided engineering support for satellite and Internet networks. Dr. Kat holds a Ph.D. degree in computer science from the Ben-Gurion University of the Negev.

Bryan S. Langston *IBM Research - Almaden, San Jose, CA 95120 USA (bryanlan@us.ibm.com).* Mr. Langston is a solutions architect in the Storage Services Research Department. He received a B.S. degree in business with an emphasis in information systems from San Diego State University in 1998 and an M.S. degree in engineering management from the University of Colorado at Boulder in 2010. He joined IBM Global Services in 1999 at the Almaden Research Center. He received the IBM Global Services 2005 Leadership Excellence Award for his team leadership with the IBM WebFountain project from 2001 to 2006. In 2006, he joined the Research Division and created and led a project focused on a new desktop management solution, which became an IBM service offering globally. He was the chief solution architect for the Smarter Traveler project with University of California–Berkeley and the California Department of Transportation. Mr. Langston is also an alumnus of the 2011 Smarter Cities Challenge Edmonton team.

Nagapramod S. Mandagere *IBM Research - Almaden, San Jose, CA 95120 USA (pramod@us.ibm.com).* Mr. Mandagere is a Senior Software Engineer in the Storage Services Research Department. After completing his bachelor's degree in 2003 from India, Mr. Mandagere received a master's degree in electrical and computer engineering from the University of Wyoming in 2005, following which he received a master's degree in computer science from the University of Minnesota in 2008. He joined IBM Research - Almaden in 2008, where he worked on Storage Systems Resiliency Management while simultaneously pursuing his Ph.D. degree at the University of Minnesota. He has authored or coauthored eight technical papers and 20 patents in the domain of systems research.

Dietmar Noll *IBM Germany Research and Development Lab, Hechtsheimer Str. 2, 55131 Mainz, Germany (dnoll@de.ibm.com).*

Mr. Noll is a Senior Technical Staff Member in Tivoli storage software development and an architect for the IBM SmartCloud Virtual Storage Center and Tivoli Storage Productivity Center products. He has worked in the area of storage management software for almost ten years with a focus on cloud storage in the recent three years. Mr. Noll is member of The Open Group and member of the German national body for the ISO/IEC JTC1/SC38 (International Organization for Standardization/ International Electrotechnical Commission, Joint Technical Committee 1) distributed application platforms and services—web services, service-oriented architecture, and cloud computing).

Sumant Padbidri *Cloud & Smarter Infrastructure Division, Almaden Research Center, San Jose, CA 95120 USA (spadbidr@us.ibm.com).* Mr. Padbidri is a Senior Technical Staff Member in the Storage Department in the Cloud & Smarter Infrastructure Division and is responsible for Storage Management strategy. He received a B.S. degree in electronics from Osmania University in 1987, and an M.S. degree in computer science from the University of Bombay in 1989. He joined IBM in 1998, where he has worked on a variety of projects. He has worked in the Storage Department since 2001. He has received multiple IBM Outstanding Technical Achievement Awards and is an author or coauthor of six patents.

Ramani Routray *IBM Research - Almaden, San Jose, CA 95120 USA (routrayr@us.ibm.com).* Mr. Routray is a Senior Technical Staff Member and Manager in the Storage Services Research Department at IBM Almaden Research Center. He joined IBM in 1998 and has architected and delivered several key products and service offerings in the area of storage Disaster Recovery, Storage Management and Virtualization, and Cloud Service Architectures. He received a bachelor's degree in computer science from Bangalore University and a master's degree in computer science from the Illinois Institute of Technology. He has received multiple IBM Outstanding Technical Achievement Awards, authored over 30 scientific publications and technical standards, and is author or coauthor of 35 patents.

Yang Song *IBM Research - Almaden, San Jose, CA 95120 USA (yangsong@us.ibm.com).* Dr. Song is a Research Staff Member in the Storage Services Research Department. His research interests include stochastic modeling and optimization, algorithm design, statistical analysis, machine learning, and their applications in data center management ranging from storage resources to networking and virtualized compute environments. He has served on the organizing and technical program committees of many conferences. Prior to joining IBM Research, he received his Ph.D. degree in the University of Florida in 2010.

Chung-hao Tan *IBM Research - Almaden, San Jose, CA 95120 USA (chungtan@us.ibm.com).* Mr. Tan is a Senior Software Engineer in the Storage Services Research Department. He received a B.S. degree in computer science from the National Taiwan University in 1997 and an M.S. degree in computer science from the University of Southern California in 2000. He subsequently joined IBM at the Almaden Research Center, where he has worked in storage management research. He has received four IBM Outstanding Technical Achievement Awards for his work on the IBM storage management products. He is also author or coauthor of several patents and technical papers.

Avishay Traeger *IBM Research Division, IBM Haifa Labs, Haifa University Campus, Mount Carmel, Haifa 31905, Israel (avishay@il.ibm.com).* Dr. Traeger is a Research Staff Member at the Storage Systems Group at IBM Research—Haifa. He is leading research and development of advanced storage features for virtualization and cloud environments, including development of storage features for cloud computing. His additional work includes data reduction and performance benchmarking and optimization with a more recent focus on virtualized I/O. Dr. Traeger holds a Ph.D. degree in computer science from the Stony Brook University. He is a core contributor for the OpenStack project.