

# Automated planners for storage provisioning and disaster recovery

S. Gopisetty  
E. Butler  
S. Jaquet  
M. Korupolu  
T. K. Nayak  
R. Routray  
M. Seaman  
A. Singh  
C.-H. Tan  
S. Uttamchandani  
A. Verma

*Introducing an application into a data center involves complex interrelated decision-making for the placement of data (where to store it) and resiliency in the event of a disaster (how to protect it). Automated planners can assist administrators in making intelligent placement and resiliency decisions when provisioning for both new and existing applications. Such planners take advantage of recent improvements in storage resource management and provide guided recommendations based on monitored performance data and storage models. For example, the IBM Provisioning Planner provides intelligent decision-making for the steps involved in allocating and assigning storage for workloads. It involves planning for the number, size, and location of volumes on the basis of workload performance requirements and hierarchical constraints, planning for the appropriate number of paths, and enabling access to volumes using zoning, masking, and mapping. The IBM Disaster Recovery (DR) Planner enables administrators to choose and deploy appropriate replication technologies spanning servers, the network, and storage volumes to provide resiliency to the provisioned application. The DR Planner begins with a list of high-level application DR requirements and creates an integrated plan that is optimized on criteria such as cost and solution homogeneity. The Planner deploys the selected plan using orchestrators that are responsible for failover and failback.*

## Introduction

With the continued growth toward petascale storage requirements, enterprise storage area networks (SANs) are increasing in size and complexity. Customers are continually purchasing and adding new storage subsystems, fabric switches, and servers. Management of these complex interconnected environments is a challenging task and is turning out to be the primary cost component in most data centers, often significantly more than hardware costs. With the increasing size and device types, several common tasks that were hitherto performed manually as an art, such as provisioning storage for workloads and disaster recovery (DR) planning, have become increasingly complex for administrators. Manual planning tends to be slow, expensive, and error prone and does not scale. Furthermore, until recently there were no suitable SAN resource management tools that were capable of collecting all of the necessary configuration and performance numbers from the various storage

components into one place to facilitate automated planning and analysis. Hence, much of the planning work was done either by expert consultants or by extensive over-provisioning, both of which are expensive and intolerable as storage demands and costs rise.

Recent developments in SAN monitoring tools have greatly advanced the state of the art. Products such as the IBM TotalStorage\* Productivity Center (TPC) [1] and EMC ControlCenter\*\* [2] provide a good foundation by monitoring SAN components and gathering the configuration and dynamic performance information in one common place with centralized control. By leveraging these advances in SAN management tools, we present two advanced planning tools that assist administrators in one of their most important, yet highly complex, tasks of application provisioning.

Introducing a new application into a data center SAN is often a multiple-week activity requiring significant manual effort. A lot of complexity is introduced by

©Copyright 2008 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied by any means or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/08/\$5.00 © 2008 IBM

storage provisioning, its connectivity to application servers, and resiliency planning for the applications. Beginning with the basic storage capacity requirement, administrators have to decide how many volumes to create and what the individual sizes should be. Then, understanding the nature of the workload performance requirements—such as the number of I/O (input/output) operations per second (iops), average transfer size, read–write ratio, random–sequential mix, and response time—administrators have to choose where to place the volumes so there are no adverse affects on existing workloads or the new storage. This is a challenging task, as modern storage subsystems have sophisticated internal structures with interlinked components and bottlenecks can occur in multiple locations, including ranks, pools, device adapters, host adapters, and processor complexes. An overloaded device adapter, for example, will adversely affect the performance of workloads in all the pools under it. Therefore, care must be taken in allocating new volumes so that none of the components in the hierarchy is overloaded. Such hierarchical constraints and the space–performance imbalances that exist among the pools and workloads make this a highly challenging task. Most of the existing work [3, 4] does not take such hierarchical constraints into consideration.

Once the volumes are created, the next step in the provisioning process is to set up the requisite number of access paths from the volumes to the hosts. This number depends on the level of redundancy desired, the I/O rate expected, and any options for supporting multiple paths between each host and storage subsystem that are available on each of the hosts. Once the paths and port pairs have been determined, the next step is to create appropriate zones and zone sets to enable the ports on the hosts to access the newly created volumes.

Next, the administrators must protect the provisioned application from disastrous events such as subsystem failures, configuration errors, or a complete site failure. With the increasing automation of business processes and the desire for continuous availability of enterprise applications, planning for disasters has become an economic necessity as well as a regulatory requirement for most enterprises. DR planning for enterprise applications requires coordination among multiple layers of hardware and software, namely virtual machines (VMs), servers, databases, file systems, IP (Internet Protocol) networks, Fibre Channel fabrics, and storage controllers. Based on rules of thumb and heuristics involving teams of administrators and consultants with expertise in different layers, the goal is to decide which replication technologies to use, the number of replicas, and the resource allocation for these replicas. Manual planning becomes highly error prone and suboptimal with increasing storage capacities, increasing heterogeneity of

device protocols, growth in the number of applications and associated policies, and an ever-growing list of replication technology choices with overlapping functionalities available at the VM [5, 6], server, database [7, 8], file system [9], and storage [10–12] levels.

Administrators need tools that can take application-level DR requirements as input and generate a DR plan with details of the replication technologies, their layer of operation, and the associated primary and secondary devices. Existing research in this domain does not address this need; the focus has been on the subproblem of selecting replication technologies within an individual tier [13–15].

In this paper, we present intelligent placement and resiliency planners that aim to assist the administrator in provisioning and protecting application storage. We describe the IBM Provisioning Planner, which identifies appropriate locations for placing newly allocated volumes. These are based on careful analysis of the current utilization of the various subsystem components and their suitability to serve the new workload considering the hierarchical constraints and the space–performance imbalances among the pools and workloads. The Provisioning Planner also plans and sets up paths between application servers and storage volumes, avoiding port bottlenecks and improving resiliency. We then present the details of the integrated DR planning framework. Our current scope for end-to-end planning is across multiple layers—storage systems, databases, servers, and VMs. In contrast to individual layer planners, the integrated planner analyzes tradeoffs between overlapping protocols at different layers and it generates DR plans with combinations of technologies across layers, which is typically the case in real-world deployments. Even within a particular layer, we differentiate between similar replication protocols provided by different vendors as they exhibit different properties for resource usage and failure recovery.

## Provisioning Planner

The Provisioning Planner is comprised of three planning modules: volume, path, and zone. Each module works in combination with the others to create a provisioning plan.

The Volume Planner takes as input the space (in gigabytes) and performance requirements (e.g., I/O demand, read–write ratio, and response time) of the new workload and recommends the number and sizes of the new volumes to be created as well as their locations. These are based on careful analysis of the current utilizations of the various subsystem components and their suitability to serve the new workload considering the hierarchical constraints and the space–performance imbalances among the pools and workloads. The architecture and design considerations of the Volume

Planner are given in the next section. The answers are provided as a recommendation to the administrator, who can accept or refine it further. Once accepted, the volume can either be created immediately or scheduled for later creation.

The Path Planner takes as input the hosts and volumes to which access must be provided and the number of desired paths between them. To create multiple paths, the multiple-path usage mode can be specified. Additional redundancy can be added by selecting the fully redundant paths option, which ensures that the paths are divided evenly across the available SAN fabrics. The Path Planner also analyzes the port utilization on both the host and the volume sides to select suitable port pairs to serve as paths. In order to avoid port bottlenecks, the port selection automatically favors those that are on the same fabric and are not already carrying a high level of other I/O traffic, which can hurt the workload performance.

The Zone Planner takes the new portpairs as input, compares them with the existing zone sets, and recommends suitable modifications, either changing existing zone sets or creating new zone sets to incorporate the new portpairs. Zoning is a security feature that uses the grouping of ports to both divide the SAN into smaller segments and to restrict which ports on hosts can access a port or set of ports on subsystems. This grouping of ports is usually referred to as *soft zoning*. The policies used by the Zone Planner and its methodology are described in subsequent sections.

The three planners—volume, path, and zone—have been implemented and are available as part of the IBM TPC V3.3 product line [1] and can be invoked either independently or in conjunction with one another. In the latter case, the output from one serves as input to the next.

### Volume Planner

In this section we describe the architecture and algorithms used in the Volume Planner. It takes as input the following workload requirements:

- Amount of **storage space required** (GB).
- **I/O demand** (iops/GB).
- **Average transfer size** (KB/s).
- **Desired RAID** (Redundant Array of Independent Disks) **level**.
- **Sequential-random mix** (percentage of each).
- **Read-write mix** (percentage of each).
- **Cache hit rate**.
- **Peak activity time**.
- **Minimum and maximum volume sizes**.

Given this input, the Volume Planner recommends the number and sizes of new volumes as well as the locations

at which the new volumes should be created while avoiding overloading any subsystem component based on current utilization. Balancing the load in this manner keeps the system predictable, ensures good response time for applications, and increases the tolerance to unexpected surges in workload demands.

Workload parameters can be entered directly or via a workload template profile. These profiles allow for the estimation of workload requirements when data is not available or to use more generic workloads. The five predefined workload template profiles are as follows:

1. *OLTP (online transaction processing) standard*—for typical OLTP.
2. *OLTP high*—for very active OLTP.
3. *Batch sequential*—for batch applications involving large volumes of data.
4. *Data warehouse*—for applications with inquiries into large data warehouses.
5. *Document archival*—for document archival applications.

The majority of application workloads typically falls into one of these categories, so the administrator can choose the one that closely matches his workload and can further modify or scale one or more of its parameters. Thus, workload profile numbers can be obtained either as templates, as direct inputs, as predictions from other workloads, or as a combination of the three.

### Nominal costs and device models

An important concept in the Volume Planner is that of nominal cost for a workload on a component. *Nominal cost* is the cost (i.e., utilization) that is incurred on the component if 1 GB of the new workload were to go through it. The component can be a storage pool at the leaf level, a device adapter, a host adapter, or a processor complex at a hierarchy level.

The nominal cost, as expected, depends on the device, the component capabilities (e.g., maximum sequential read, maximum sequential write, and maximum random read I/Os that can be handled), and the workload parameters (e.g., iops, transfer size, read-write ratio, and random-sequential ratio).

These nominal costs can be obtained using Disk-Magic [16]-like white-box models (only internals can be viewed), black-box models (only inputs and outputs can be viewed), or device-specific calculations. The rest of the Volume Planner is unaffected by how these nominal costs are obtained, so any preferred available method can be used.

Black-box models have also been studied for capturing storage subsystem behavior. As suitable, reliable ones

become available now or in the near future, they can also be used to estimate component utilization for the workloads.

For scenarios in which calling such device models in an online fashion is not possible, a third, potentially less accurate alternative is to infer utilization on the basis of simple device capability calculations. For example, for a device adapter or host adapter, given that

$$io-util = rriops/max\_rriops + rwiops/max\_rwiops \\ + (srmbps + swmbps)/(SSSZ*max\_rriops)$$

and

$$txfr-util = (srmbps + rrmbps)/max\_srmbps \\ + (swmbps + rwmbps)/max\_swmbps,$$

then

$$nominal-cost = \max\{io-util, txfr-util\},$$

where the prefixes *rr*, *sr*, *rw*, and *sw* refer to random read, sequential read, random write, and sequential write, respectively; the suffixes *iops* and *mbps* refer to the corresponding I/O operations per second and megabits per second, respectively; *SSSZ* refers to the sector size; the numerators (e.g., *rriops*) refer to the current load along that measure (e.g., number of random read I/O operations per second), while the denominators (e.g., *max\_rriops*) refer to the maximum that can be supported along that measure; *io-util* and *txfr-util* capture the extent to which the device is loaded in terms of the number of I/O operations it can support per second and the number of megabits it can transfer per second, respectively. However the nominal costs are obtained—white box, black box, or calculations—the rest of the Volume Planner process is orthogonal to this. The main Volume Planner algorithm that determines the number and sizes of volumes needed and their load-balanced allocations on the basis of current utilizations is described in the next section. It assumes that the nominal costs are available.

#### ***Selection process: Configuration phase***

In the configuration phase, the Volume Planner retrieves the SAN configuration data from the database, the list of available subsystems pruned by the user preferences, and then eliminates storage pools that do not have adequate available capacity to meet the minimum volume size requirements. It then computes the current utilization levels for each component on the basis of statistics collected by the monitoring engine.

#### ***Selection process: Load-balancing phase***

The Volume Planner uses the basic premise behind the worst-fit approach but performs volume allocation in a sophisticated manner (as described in next section) in

order to provide load-balancing at different levels of the resource hierarchy.

The regular ***first-fit approach*** for memory management selects the first available pool for allocating a new volume, thus reducing allocation complexity at the expense of performance. The ***best-fit approach*** attempts to minimize fragmentation by selecting the most utilized component that can fit a new volume. Once the most-utilized pool is chosen, volumes are allocated to this pool until it is depleted. While this strategy minimizes fragmentation, it does not balance the load. The ***worst-fit approach***, on the other hand, uses the least-utilized component to allocate new volumes. This strategy attempts to minimize local regions of high activity. However, not one of these approaches by itself addresses the ***space-performance tradeoffs*** that exist between storage pools or addresses the hierarchical bottlenecks and utilization numbers at higher levels of the hierarchy.

#### ***Volume allocation algorithm***

The volume allocation algorithm addresses challenges of space-performance tradeoffs among storage pools and the hierarchical bottlenecks and utilization numbers at higher levels of the hierarchy. The algorithm input parameters are required space (in gigabytes), minimum volume size, maximum volume size, workload profile, and resource graph.

The volume allocation algorithm is composed of two stages: calculating the usable space at each node and performing storage allocation. Usable space at each node is a single number that captures the space and performance utilization of the node and its hierarchy. This metric is calculated using a three-step process:

1. For each node, determine how much new workload (in gigabytes) can be put on the node locally before its performance utilization exceeds the target percentage.
2. Adjust for the parent limitation of each node in a top-down, depth-first manner by setting the node usable space metric to the parent metric when the parent is smaller.
3. Adjust for the child limitations upward to the parent limitations in a depth-first manner by the parent usable space metric to that of the sum of its children when the child sum is greater than the parent.

The resulting usable space metrics are used to filter out all possible targets that do not meet the minimal threshold requirements and to select the best targets from the remaining targets. After filtering, the remaining pools are sorted in decreasing order on the basis of the usable metric. Suitable volumes are then found by traversing this



list and finding the volumes with parameters that meet the performance and size requirements, giving selection weight to the pools higher in the list. Additional details can be found in Reference [17].

### ***Other related considerations in volume planning***

#### ***Unassigned volumes***

Often there may be volumes in the system that are not currently being used for any workload. This could be either because the workload for which they were created has been removed but the corresponding volumes have not been removed or because an administrator has created volumes in advance. The Volume Planner gives an option to reuse such existing unassigned volumes. If selected, the Volume Planner first checks whether unassigned volumes exist in the current selected pool. If so, it recommends them before recommending creation of a new volume in that pool.

#### ***Space-only mode***

Along with the five predefined templates for workload performance profiles, the Volume Planner provides a sixth fallback option: space-only mode. In this mode, the volume recommendations are based only on the space requirement. Intended primarily for backward compatibility for administrators familiar with space-only-based allocations, this is also useful for over-provisioned scenarios in which performance is not a critical concern. The minimum volume size and maximum volume size requirements will still exist, and the Volume Planner implements this mode by setting nominal costs to zero or to very small numbers.

### ***Path Planner***

In this section, we briefly describe the Path Planner, which helps choose a suitable number of paths from the hosts to the volumes after the volumes and their locations are determined. This is dependent on the type of multipathing supported on the hosts, the amount of redundancy required, and the I/O rate desired from the hosts to the volumes. Once the number of paths is determined, the planner then helps choose port pairs on the host and subsystem that are not already carrying a high load of other traffic.

The Path Planner takes as input the list of hosts and the volumes they must access, the desired I/O rate from the hosts to the volumes, whether fully redundant paths are required, if multiple paths are to be used, and whether the mode is failover, load-balancing, or round-robin.

In *failover mode*, the multipath driver on the host uses a single path through the SAN until a failure occurs along the primary path. At that point, the driver switches to using a backup path. In *load-balancing mode*, it directs the

packets across the available paths on the basis of the load on each path and the response time to each controller. The direction strategies used depend on the specific multipath driver. In *round-robin mode*, the driver places packets on the available paths in a round-robin fashion. Although this provides a reasonably even distribution of the current workload among the available paths, it does not take into account the load already existing on the paths from other workloads and the path-dependent delays.

The Path Planner checks the capabilities of the drivers and their compatibility with the selected subsystems and the chosen multipath options. If there is a mismatch or incompatibility, it raises a warning flag. Otherwise, it selects a suitable set of host initiator and subsystem target port pairs on the same fabric to serve as paths for this volume. The pairing needs to be done carefully to ensure that the ports are not already overloaded and that there are enough pairings possible for all the hosts given their fabric connectivity.

The pairing process is composed of three components. The first two analyze the port and connectivity data in the storage resource manager database to calculate a port load metric for each port and assign a host ranking. These metrics, along with the list of host and subsystem ports, serve as input to the port pairing process to produce the datapaths.

**Port load metrics**—A load metric is calculated for each host and subsystem port on the basis of the number of I/O connections and the relative amount of total traffic each port is supporting. The necessary input data for this calculation comes from the TPC storage management database. When no performance data is available, the load metric is calculated using only the I/O connection information.

**Host ranking**—The hosts are ranked so that the hosts with the least number of available storage controller paths are processed first. The number of storage subsystem ports available to each host port is found on the basis of the fabric it uses and is then summed for each host. The host list is ordered according to the ranking so that the hosts with the lower number of possible subsystem ports are toward the top of the list.

**Port pairing**—Datapaths are constructed by pairing host and subsystem ports to ports with the least load on the basis of the port load metrics. Starting with the first host in the host ranked list, the host port with the lowest metric is found. Next, the storage subsystem port with the lowest metric that is attached to the same fabric as the host port is found. The resulting datapath port pair is added to the datapath list. The load metric for each port is adjusted to reflect the added traffic from this datapath. The process is repeated for the next host.

Once the pairing process has created one datapath for each host, the next datapath is constructed. When the fully redundant path policy is selected, the port-pairing process selects the host initiator port from the set of ports that are members of fabrics that are different than the first host initiator port; otherwise, the next host initiator port will be selected only on the basis of the port load metric. When this round of port pairing is complete but the number of datapaths for each host is not yet complete, the pairing process continues until the required number of datapaths has been created.

The final output is composed of the target initiator port pairs that determine the paths between the hosts and storage subsystem.

### Zone Planner

Once the volumes have been created and the paths and port pairs are determined, the next step is to set up zones to enable the hosts to access the volumes.

Zoning is a security feature in SANs that limits which host ports can access which one or more storage subsystem ports. Only host–subsystem port pairs that occur together in the same zone are allowed to access each other. Thus, when new storage is being provisioned, the host initiator port and the target subsystem port should be zoned together by either creating a new zone or updating an existing zone. To do this, the zones and zone sets that are already in use and the zoning guidance policies or validation policies that are in effect must be determined.

The Zone Planner helps to determine the appropriate zone changes. The inputs to the Zone Planner include the new host–subsystem port pairs that must be zoned together, and the guidance and validation policies (e.g., maximum number of zones and maximum number of zone members per zone) that are in effect.

*Guidance and validation policies*—A guidance policy can be selected from among the following:

- One zone per host bus adapter (HBA).
- One zone per host.
- One zone per group of hosts that run a single application.
- One zone per storage controller.
- One zone per storage controller type.

Sample validation policies include the following:

- Two HBAs from different vendors should not be in the same zone.
- Two controllers of different types should not be in the same zone.
- The maximum number of members in a zone =  $N_m$  (parametric policy).

- The maximum number of zones in a fabric =  $N_z$  (parametric policy).

These policies are derived from the best practices developed by the IBM SAN Central team over several years. The primary motivation for these best practices is to avoid potential incompatibilities and problems that can arise when mixing and matching devices from different vendors. The maximum  $N_m$  and the maximum  $N_z$  are fabric policies that allow enforcing known limitations that may exist in the storage fabric.

The Zone Planner uses a two-stage process for creating zones that satisfy both the guidance and validation policies. The initial zones are created using the largest granularity in each zone that satisfies the guidance policy and the basic zoning best practices.

After the initial zones are created, they are validated against the selected validation policies. When a validation policy is violated, the zones are adjusted to satisfy the validation policy while continuing to still meet the conditions of the initial guidance policy. An alert is raised when the zoning cannot be adjusted to satisfy all the violation policies using zoning best practices.

The Zone Planner can be invoked in conjunction with the Volume Planner and Path Planner, in which case the host and subsystem port pairs are obtained directly from the Path Planner output. It can also be invoked directly by the user specifying the host ports and target ports they would like to add in the zones.

The output from the Zone Planner is the set of new zones to be created and the existing zones to be updated.

### Combined orchestration

When the three provisioning planning modules are invoked in conjunction with one another, the output from one becomes input to the next.

The planners have been implemented and are available as part of the IBM TPC V3.3 product line. See [18] for additional details and user interface screens. If the planners are invoked together, the result is the set of volumes to be created and the set of zone changes to enable the hosts to access those new volumes. The administrator can choose to accept the recommendations or refine them further. If accepted, the TPC storage management tool can be invoked to execute those recommendations by creating the new volumes and setting up the zones. The administrator can choose whether to do it immediately or schedule it for a later time.

### DR Planner

Critical enterprise application downtime can translate into millions of dollars in losses for the enterprise [19]. Typically, at the time of provisioning resources for an

application, administrators want to ensure protection for disasters such as site failure, server failure, storage subsystem failure, virus failure, and configuration errors. The planning for application-level DR includes multiple layers, namely VMs, servers, networks, and storage subsystems, with each layer supporting one or more replication technologies that overlap in functionality with other layers. Given an administrator-specified list of application-level DR requirements, it is a nontrivial task to generate a plan of replication technology configurations at each layer. Further, deployment of such a plan would require expertise in multiple technologies. In this section, we describe the DR Planner, which aims to address these issues by automating the process of end-to-end DR planning and deployment. We start with key terminology and a brief background of DR technologies, followed by the design details of the DR Planner.

### **DR terminology and background**

In describing the details of the DR Planner, we use terminology commonly used in DR planning. DR requirements are specified for a *data source*, which is a logical entity representing an application, database, or file system that requires protection against disasters. The requirements are specified as one or more *profiles*, where each profile specifies the failure protection type (site, subsystem, virus) and the corresponding values of the recovery point objective (RPO), recovery time objective (RTO), and the application impact. RPO is expressed in seconds or minutes and corresponds to the loss of updates the user is willing to tolerate in the event of a failure; it represents how quickly the updates are propagated from the primary to the secondary data sources. The RTO, expressed in seconds or minutes, corresponds to the system downtime; major online portals usually have RTOs of less than a minute. Application impact corresponds to the added latency, in milliseconds, to the application due to data replication. The term *failover* refers to the recovery after a failure—switching from the primary copy to the secondary copy.

### **DR technologies**

A number of replication technologies are available from various vendors at multiple layers in the invocation path: VMs, physical servers, databases, file systems, appliance gateways, and storage controllers. In this section, we briefly describe some of the most prevalent technologies.

Server failover technologies include server clustering solutions, such as the Symantec Veritas\*\* Cluster Server (VCS) and IBM High Availability Cluster Multiprocessing (HACMP), that use a heartbeat mechanism (which periodically polls whether the servers are operational) to detect server failures and, in response, migrate applications from a failing node to a healthy

node. Server virtualization technologies such as VMware [20] and Xen [21] also provide application VM migration mechanisms. Recent developments in VM backup technology offer great opportunities to advance the state of the art. One such development is the VMware VM Snapshot technology [20]. The VM snapshot captures the entire state of the VM at the time of the snapshot, including the state of all the VM disks, the contents of the VM memory, and the VM settings. By reverting to an existing snapshot, the application can return to the same state as that at the time of the snapshot. This provides significant RTO advantages, as the application does not require restarting after a failure. Additionally, this technology is useful to protect against accidental VM deletions.

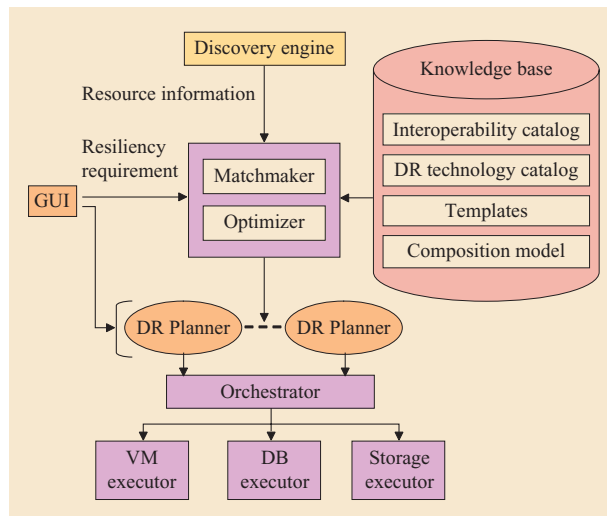
Storage replication technologies can be broadly classified into synchronous or asynchronous replication and volume duplication by means of split-mirror or copy-on-write (point-in-time) techniques [22]. Synchronous replication ensures that each write to disk is immediately copied to the secondary site. This ensures zero data loss in the event of a failure (zero RPO), but at the cost of high application performance impact. With asynchronous replication, write completions are returned to the application once they have been committed to the primary disk. Updates on the secondary volume are performed at a later time. This is useful for long-distance replication, but the RPO and RTO may be significant. Split-mirror or point-in-time replication provides an instantaneous copy of a storage volume with minimal impact on the application. It is useful for preserving the point-in-time images at different time instants and is used to recover from human or application errors or failures caused by viruses.

### **DR Planner**

**Figure 1** shows the architecture of the DR Planner. We briefly introduce the key modules with details in the following sections.

*Discovery engine*—Finds the servers, network, and storage devices associated with applications and gathers static device configuration, interconnectivity data, dynamic performance statistics, and event logs. Additionally, it collects configuration information about databases and other installed software. There are several management offerings such as IBM TPC [1] and the EMC ControlCenter [2] that automate discovery and monitoring of the information technology infrastructure. DR Planner uses IBM TPC for discovery.

*Knowledge base*—Consists of information defined by the system designer and administrator for best practices and replication technologies. The recipe templates express well-known replication technology configurations corresponding to a certain set of DR requirements. The



**Figure 1**

The Disaster Recovery (DR) Planner architecture. (GUI: graphical user interface; VM: virtual machine.)

technology catalog lists canonical models of available replication technologies from various vendors that operate at the VM, database, and storage controller tiers. For each replication technology, the catalog defines the technology class, DR specifications (RPO, RTO, and application impact), resource usage models in terms of the CPU (central processing unit), I/O, and network as a function of the load characteristics, and protocol taxonomy (in terms of fault coverage, copy divergence, propagation order, and acknowledgment). Additionally, the knowledge base also defines interoperability constraints and composition constraints of technologies for which the target volume of one replication configuration serves as the source for the other.

**Matchmaker module**—Responsible for finding combinations of one or more replication technologies that can be used to satisfy the DR requirements specified by the administrator. The Matchmaker module finds options using the best-practice templates as well as composition of technologies from the catalog.

**Optimizer**—Finds a feasible solution that can be deployed given the following:

- The DR options generated by the Matchmaker for one or more enterprise applications.
- The RPO, RTO, and resource usage properties of each solution.
- The available resource usage.
- Administrator-defined priorities and objective metrics such as cost, application latency impact, and homogeneity of technologies.

**Runtime orchestrator**—Technologies across multiple layers need to synchronize during normal operation (e.g., a storage point-in-time copy needs to be synchronized with a freeze of operations at the VM level) as well as during failover (e.g., restart an application, first restart storage, then VM, followed by the application). Most replication technologies have scripts and commands for failover and periodic synchronization; DR Planner orchestrates their invocation using predefined work flows for technology combinations at different tiers.

### Knowledge base

The ability to find replication options in DR Planner is only as good as the information in its knowledge base. Ideally the knowledge base should be continuously updated by vendors releasing new technologies. Best-practice templates are typically obtained from case studies, product documentation, and analysis of deployed solutions.

### Best practices

Best practices capture the inherent knowledge of a DR expert. In DR Planner, there are four different types of best practices: interoperability constraints, functionality class templates, solution templates, and orchestration workflows.

**Interoperability constraints**—A prerequisite table of version and vendor details for instantiating a particular replication technology including server, fabric, and storage controller interoperability. It also defines prerequisites for automation technologies (e.g., cascading, where the secondary copy of technology A serves as a primary for technology B).

**Functionality template**—Maps the DR requirement, such as site protection, to a set of one or more technology functions that need to be supported by the replication configuration. The DR Planner prototype maps the DR requirements to the following functionality classes: server-level clustering, VM snapshot facility, database snapshot facility, synchronous database copy facility, asynchronous database copy facility, storage flash copy, synchronous storage copy, and asynchronous storage copy.

**Solutions template**—A recipe to satisfy two or more DR profiles. Each template defines details of the recipe in terms of the number of copies, the replication technology, and any restrictions in terms of cascading replication technologies. A commonly used solution template is a synchronous replication followed by an asynchronous replication. This template provides site protection across intercontinental distances as well as site protection within 300 km at the same time. Further, the secondary copy has an RPO of zero.



**Orchestration templates**—Consist of workflows for synchronization and failover operation across technologies that exist at different levels. The templates define sequences of do and undo and freeze and resume operations for combinations such as VM snapshot with database snapshot. Additional details are described in the section on orchestration.

### Replication technology catalog

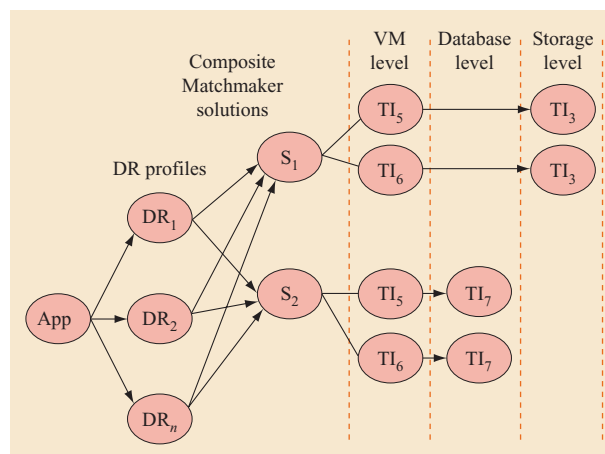
The replication technology catalog consists of canonical models for replication technologies that capture information required for the planning process. A replication technology is a tuple of  $\langle \text{ServiceClass}, \text{Resource} \rangle$ . *Resource* represents formulas that predict the host CPU usage in cycles per second, IP network usage, and storage utilization as a result of replication setup for given application workload characteristics. The parameters in the application workload parameters are the write rate and the percentage of in-place rate during a specified time window.

*ServiceClass* is a triplet of functionality, layer, and properties. *Functionality* defines the operation of the replication technology, such as database snapshot or storage flash copy. *Layer* is the operation level, namely VM, server, database, or storage. *Properties* are represented by four parameters: RTO, RPO, latency impact, and distance constraint (in miles). The property values are typically available in product documentation for the replication technology.

### Matchmaker module

For each business-critical application, the administrator specifies requirements as one or more DR profiles. The Matchmaker uses these profiles as input to search the knowledge base for replication technologies that can be applied to satisfy the requirements. Formally stated, the input to the Matchmaker is a set of DR profiles for an application:  $\text{DRApp} = \{\text{DR}_1, \text{DR}_2, \dots, \text{DR}_n\}$ . The output as shown in **Figure 2** is a set of candidate solutions in which each solution is either a single replication technology instance (TI) or a combination of instances at the VM, database, or storage levels:  $\text{SolApp} = \{ \langle \text{TI}_5, \text{TI}_7 \rangle, \langle \text{TI}_6, \text{TI}_7 \rangle, \langle \text{TI}_5, \text{TI}_3 \rangle, \langle \text{TI}_6, \text{TI}_3 \rangle \}$ .

To find replication technology options, the Matchmaker uses a combination of two searches: either searching solution templates for popular DR solutions that can match the specified DR requirements or using functionality templates to map DR profiles to technology requirements and searching the technology catalog. The outcome of the two searches is consolidated in *SolApp*. Solution templates have an advantage in mission-critical applications in which enterprises may not be willing to experiment with new combinations of replication technologies, thus preferring a template-based approach.



**Figure 2**

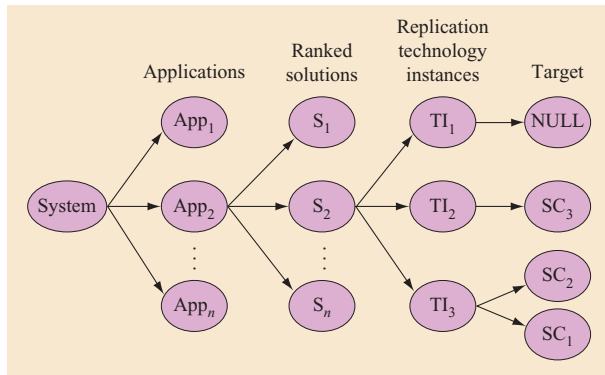
Output of the Matchmaker module.

The disadvantage of the template-based approach is that there are fewer solution choices, especially if the DR requirements of an enterprise are uncommon. The template-based approach restricts itself to the expert-codified combination of solutions (which in many cases is a very small portion of the search space).

The matchmaking process can be described using four broad scenarios:

1. The DR requirements of the application consist of a single DR profile (e.g., virus protection only) that matches a template or technology in the knowledge base.
2. The DR requirements consist of more than one DR profile that is satisfied by a single template or technology. Another variation is that each DR profile is independently satisfied by the technology, in which case the solution is a combination of more than one technology.
3. Individual DR requirements (such as site-level protection) can be satisfied only by using a combination of technologies.
4. One or more DR requirements do not match with any template or technology, resulting in the planning process raising an alert for human intervention.

For cases 1 and 2, the Matchmaker instantiates results from solution templates and the catalog. For case 3, matchmaking is more involved, requiring calculation of DR properties of a composite technology that given the canonical models of individual technologies is nontrivial. DR Planner uses inductive composition logic to solve this problem. The problem of composition can be formally



**Figure 3**

The optimization search space.

stated as follows: Given the canonical models of two technologies A and B, predict the *ServiceClass* and *Resource* for the composite technology of A and B. The composition can be either a sequence of A and B ( $A \rightarrow B$ ), where A is the primary copy of technology B, or A and B in parallel ( $A||B$ ), where the primary copy for technology A is also the primary copy of technology B. Approximating *Resource* for the composition is fairly straightforward since it is an additive function. Similarly, among the *ServiceClass* parameters, latency is additive, but RPO and RTO are not simple to predict because each requires understanding the protocol details for A and B. The rest of the section describes details of inductive composition logic to address this problem.

A simple representation of composition logic is to have formulas for all possible technology combinations. For example, consider the composition of synchronous data replication using metro mirror (MM) with asynchronous long-distance global mirror (GM). As shown in

$$\text{RTO}(\text{MM} \rightarrow \text{GM}) = \text{RTO}(\text{GM}) + \Delta$$

and

$$\text{RPO}(\text{MM} \rightarrow \text{GM}) = \text{RPO}(\text{GM}) + \text{RPO}(\text{GM}),$$

the formulas are derived by observing that the recovery step consists of putting the target copy of GM online and making it accessible to clients. Hence, the recovery time equals  $\text{RTO}(\text{GM})$  and a  $\Delta$ , where  $\Delta$  captures the time it takes for changing the routing table. Similarly, the formula for RPO is based on the observation that data staleness is added along the sequence. The limitation of this approach is the large number of compositional formulas that have to be created beforehand.

In inductive composition logic, formulas are defined for categories of replication technologies and framed in an inductive manner where point B (or single-copy)

replication technology is attached either in sequence or in parallel with a composite replication technology A. The technology categories are similar to those for functionality templates. The formulas given to DR Planner are configurable and based on a detailed study of the replication technologies.

### Optimization module

An enterprise typically has multiple applications that are important to its business. Given their individual DR requirements, the Matchmaker generates a set of solutions for each application. The goal of the Optimizer module is to instantiate the best option for each application on the basis of its priority, ensuring that the overall objective function (such as cost, RPO, or homogeneity) is optimized. **Figure 3** shows the search space complexity, which is NP-hard. Similar complex problems, namely the allocation of data to storage systems [23], are addressed using heuristics; the DR Planner implements the heuristic algorithm relax-rank-resolve (R3) to decide the selection of replication configuration for each application.

Assume the objective function is finding the least-cost solution that minimizes the RPO of each application on the basis of its priority. The R3 algorithm begins by finding the candidate secondary devices to instantiate solutions for each application while creating independent partitions of applications that do not have any secondary devices in common. The intuition is that two applications that are not sharing any target servers, storage, or fabric can therefore be independently optimized. The partitioning involves instantiating solutions and determining the potential target devices. For example, to instantiate synchronous data replication using peer-to-peer remote copy (PPRC) [24], the DR Planner will search for devices in the infrastructure that support the interoperability constraints defined in the knowledge base. In addition to the target devices, the partitioning also takes into account the interconnecting storage fabric and IP network between the secondary and the primary application devices. It is not uncommon for applications in the enterprise to be running on independent silos of SAN with dedicated hardware.

*Relax* refers to parameter relaxation. For applications in the same partition, we find the server, network, or storage resources that are the most constrained, that is, those having limited spare bandwidth (found using the minimum flow technique [25]) and that are labeled as the relaxation parameter used in the next step of ranking. Replication technologies typically add to the resource overheads depending on the tier of operation in the invocation stack, where each may need different types of resources, namely CPU, local and remote networks, and SAN bandwidth. Analyzing the system from the perspective of available resources helps to filter out a significant subset of solutions

for which replication may result in additional overhead on an already constrained resource.

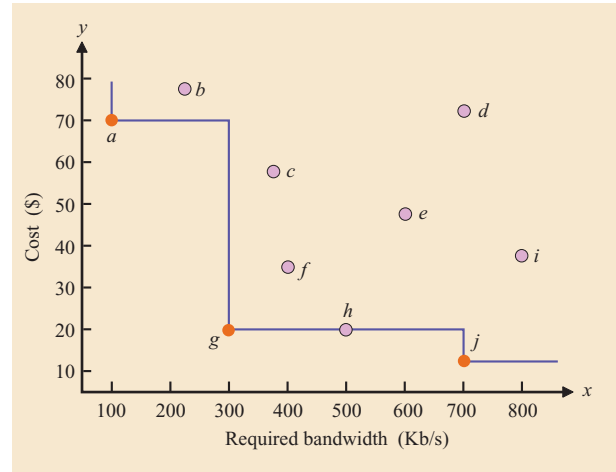
*Rank* does a two-dimensional sorting of the solutions for each application, where the two dimensions are the relaxation parameter (e.g., intersite network bandwidth) and the objective parameter (e.g., cost). The goal is to find a solution with the lowest cost in combination with the lowest use of network bandwidth. DR Planner uses the Skyline algorithm [26] of information retrieval to find such solutions. An example of the Skyline approach is shown in **Figure 4**, which shows the ranking of plans labeled from *a* to *j*. In the example, if the available bandwidth was 300 Kb/s, the ranking would select plan *g* as the highest ranked plan.

*Resolve* is a greedy bin-packing process whose goal is to minimize  $\sum_{i=0}^{App} P_i \cdot RPO_i$ , where  $P_i$  is the priority of application. The algorithm picks the highest ranked solution for each application in the partition and tries it to bin-pack in the available resources. If the bin-packing fails, the second highest ranked solution (for the lowest priority application) is selected and bin-packing is retried; this process repeats until a solution is found or all the options are explored. If the bin-packing succeeds, the result along with the objective value is saved and the process can be repeated with the next relaxation parameter. **Figure 5** illustrates the working of the bin-packing formulation.

## Conclusions and related work

In this paper, we presented an end-to-end planner to assist administrators in planning and provisioning storage on the basis of capacity, workload, and resiliency requirements.

According to our survey, there is no application-based end-to-end planner that performs the planning operation on the basis of capacity, workload, and resiliency in a cohesive fashion. However, several research contributions and product offerings try to address these requirements as a piecemeal solution. Related to provisioning planning is capacity planning, which typically deals with planning for new storage infrastructures or extending existing ones. Currently, many capacity planning tools are available such as SAN Designer from Computer Associates [27], research prototypes for capacity and SAN planning from Hewlett-Packard such as Minerva [3], Ergastulum [28], Hippodrome [4], and Appia [29]. Research in the area of DR has focused mostly on replication mechanisms that allow for the efficient replication of data. Reference [12] lists a large number of replication technologies available from IBM, and various vendors provide similar functionality [30]. Azagury et al. [22] present advanced point-in-time copy technologies, and Ji et al. [31] present the mirroring replication technology Seneca that advances the state of the art in asynchronous replication



**Figure 4**

Skyline ranking for two-dimensional sorting.

```

Objective:  $\sum_{i=0}^{App} P_i \cdot RPO_i$ 
Constraints:
 $RPO_i$  is the RPO for the ranked solution for  $App_i$ 
for (solution from each App) do
    Find tuples for  $\langle CPU_i, DevID \rangle, \langle Net_i, DevID \rangle, \langle Stg_i, DevID \rangle$ 
end for
for (device in infrastructure) do
    Get average utilization in information to find tuples  $\langle CPU_{avail},$ 
     $DevID \rangle, \langle Net_{avail}, DevID \rangle, \langle Stg_{avail}, DevID \rangle$ 
end for
if Match the resource requirements of solutions to the devices then
    return objective value
else
    return -1
end if

```

**Figure 5**

Resolve bin-packing formulation algorithm.

mechanisms. Recently, the actual task of DR planning has attracted attention from researchers. Keeton et al. [13–15] present a formal framework for studying storage system dependability and propose methodologies for DR planning and details of the actual recovery process for a single application. Gaonkar et al. [32] extend that to include support for shared applications. Research is ongoing in the area of service-level-agreement-based data center management [33], and migration [34] for dynamic resource provisioning also needs to be incorporated.

For a complete autonomic data center management system, resource planning, provisioning, migration, and optimization must be done in a cohesive fashion. We believe the suite of tools described in this paper advances the state of the art.

\*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 EMC Corporation, Symantec Corporation, or Sun Microsystems in the United States, other countries, or both.

## References

- IBM Corporation, IBM TotalStorage Productivity Center Suite; see <http://www-03.ibm.com/systems/storage/software/center/index.html>.
- EMC Corporation, EMC ControlCenter Family; see <http://www.emc.com/products/family/controlcenter-family.htm>.
- G. A. Alvarez, E. Borowsky, S. Go, T. H. Romer, R. Becker-Szendy, R. Golding, A. Merchant, M. Spasojevic, A. Veitch, and J. Wilkes, "MINERVA: An Automated Resource Provisioning Tool for Large-scale Storage Systems," *ACM Trans. Comp. Syst.* **19**, No. 4, 483–518 (2001).
- E. Anderson, M. Hobbs, K. Keeton, S. Spence, M. Uysal, and A. Veitch, "Hippodrome: Running Circles Around Storage Administration," *Proceedings of the First USENIX Conference on File and Storage Technologies*, Monterey, CA, 2002, pp. 175–188.
- Vizioncore, Inc., VReplicator; see <http://www.vizioncore.com/vReplicator.html>.
- VMware, Inc., Disaster Recovery Virtualization: Protecting Production Systems Using VMware Virtual Infrastructure and Double-Take; see <http://www.vmware.com/resources/techresources/999>.
- W.-J. Chen, C. Chandrasekaran, D. Gneiting, G. Castro, P. Descovich, and T. Iwahashi, "High Availability and Scalability Guide for DB2 on Linux, UNIX, and Windows," *IBM Redbooks*, September 2007; see <http://www.redbooks.ibm.com/redbooks/pdfs/sg247363.pdf>.
- Oracle, Oracle Data Guard; see <http://www.oracle.com/technology/deploy/availability/htdocs/DataGuardOverview.html>.
- Microsoft Corporation, Distributed File System Technology Center; see <http://www.microsoft.com/windowsserver2003/technologies/storage/dfs/default.aspx>.
- EMC Corporation, EMC Replication Manager; see <http://www.emc.com/products/detail/software/replication-manager.htm>.
- C. Warwick, Ed., *IBM TotalStorage Solutions for Disaster Recovery*, IBM Press, 2004; ISBN 073849867X.
- C. Brooks, C. Leung, A. Mirza, C. Neal, Y. L. Qiu, J. Sing, F. T. H. Wong, and I. R. Wright, "IBM System Storage Business Continuity Solutions Overview," *IBM Redbooks*, February 16, 2007; see <http://www.redbooks.ibm.com/abstracts/sg246684.html?Open>.
- K. Keeton, C. Santos, D. Beyer, J. Chase, and J. Wilkes, "Designing for Disasters," *Proceedings of the Third USENIX Conference on File and Storage Technologies*, San Francisco, CA, 2004, pp. 59–62.
- K. Keeton, D. Beyer, E. Brau, A. Merchant, C. Santos, and A. Zhang, "On the Road to Recovery: Restoring Data After Disasters," *ACM SIGOPS Operating Syst. Rev.* **40**, No. 4, 235–248 (2006).
- K. Keeton and A. Merchant, "A Framework for Evaluating Storage System Dependability," *Proceedings of the 2004 International Conference on Dependable Systems and Networks*, Firenze, Italy, 2004, p. 877.
- IntelliMagic, Disk Magic; see <http://www.intellimagic.net/en/product.phtml?p=Disk+Magic>.
- S. Jaquet, M. Korupolu, K. Magoutis, K. Voruganti, and O. Zaki, *Planner for Enterprise Storage Controllers*, Version 1.0, February 2006.
- M. Lovelace, M. Defiebre, H. Gunatilaka, C. Neal, and Y. Xu, "TotalStorage Productivity Center V3.3 Update Guide," *IBM Redbooks*, December 31, 2007; see <http://www.contingencyplanningresearch.com/2001%20Survey.pdf>.
- Eagle Rock Alliance, Ltd., Online Survey Results: 2001 Cost of Downtime; see <http://www.contingencyplanningresearch.com/2001%20Survey.pdf>.
- VMware, Inc.; see <http://www.vmware.com/>.
- P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, Bolton Landing, NY, 2003, pp. 164–177.
- A. Azagury, M. E. Factor, J. Satran, and W. Micka, "Point-in-Time Copy: Yesterday, Today and Tomorrow," *Proceedings of the 10th Goddard Conference on Mass Storage Systems and Technologies/19th IEEE Symposium on Mass Storage Systems*, College Park, MD, 2002, pp. 259–270.
- J. Wolf, "The Placement Optimization Program: A Practical Solution to the Disk File Assignment Problem," *ACM SIGMETRICS Perf. Eval. Rev.* **17**, No. 1, 1–10 (1989).
- G. Castets, B. Mellish, D. Leplaideur, and M. Thordal, "IBM TotalStorage Enterprise Storage Server PPRC Extended Distance," *IBM Redbooks*, June 24, 2002; see <http://www.redbooks.ibm.com/redbooks/pdfs/sg246568.pdf>.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, Second Edition, MIT Press, Cambridge, MA, 2001, pp. 643–700; ISBN 0-262-53196-8.
- A. Singhal, "Modern Information Retrieval: A Brief Overview," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* **24**, No. 4, 35–43 (2001).
- CA, Inc., CA SAN Designer; see <http://www.ca.com/us/products/product.aspx?ID=4590>.
- E. Anderson, M. Kallahalla, S. Spence, R. Swaminathan, and Q. Wang, "Ergastulum: Quickly Finding Near-Optimal Storage System Designs," *ACM Trans. Computer Sys.* **23**, No. 4, 337–374 (2005).
- J. Ward, M. O'Sullivan, T. Shahoumian, and J. Wilkes, "Appia: Automatic Storage Area Network Fabric Design," *Proceedings of the First USENIX Conference on File and Storage Technologies*, Monterey, CA, 2002, pp. 203–217.
- A. Chervenak, V. Vellanki, and Z. Kurmas, "Protecting File Systems: A Survey of Backup Techniques," *Proceedings of the Joint NASA and IEEE Mass Storage Conference*, College Park, MD, 1998, pp. 235–242.
- M. Ji, A. Veitch, and J. Wilkes, "Seneca: Remote Mirroring Done Write," *Proceedings of the USENIX Technical Conference*, June 2003, pp. 253–268.
- S. Gaonkar, K. Keeton, A. Merchant, and W. H. Sanders, "Designing Dependable Storage Solutions for Shared Application Environments," *Proceedings of the International Conference on Dependable Systems and Networks*, Philadelphia, PA, 2006, pp. 371–382.
- K. Appleby, S. Fakhouri, L. Fong, G. Goldszmidt, M. Kalantar, S. Krishnakumar, D. P. Pazel, J. Pershing, and B. Rochwerger, "Océano—SLA Based Management of a Computing Utility," *Proceedings of the IEEE/IFIP International Symposium on Integrated Network Management*, 2001, pp. 855–868; see <http://roc.cs.berkeley.edu/294fall01/readings/oceanoIM01.pdf>.
- E. Anderson, J. Hall, J. D. Hartline, M. Hobbs, A. R. Karlin, J. Saia, R. Swaminathan, and J. Wilkes, "An Experimental Study of Data Migration Algorithms," *Proceedings of the Fifth International Workshop on Algorithm Engineering*, Aarhus, Denmark, 2001, pp. 145–158.

Received October 1, 2007; accepted for publication February 7, 2008; Internet publication June 18, 2008



**Sandeep Gopisetty** *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120 (sandeep@almaden.ibm.com)*. Mr. Gopisetty is a Senior Technical Staff Member and manager. He leads autonomic storage management research, where he is responsible for the strategy, vision, and architecture of the TPC and its analytics. He is currently working on various optimization and resiliency analytics for autonomic storage resource manager and integrated systems management. He is the recipient of several patents and IBM corporate recognition awards including an Outstanding Innovation Award and a Supplemental Outstanding Technical Achievement Award for his vision and technical contributions to the architecture of the TPC as well as leadership in driving his vision into plan and through implementation with a team that spanned three divisions. He also received an Outstanding Technical Achievement Award and a Supplemental Outstanding Technical Achievement Award, both for character recognition. His research interests include object-oriented systems, Sun Java\*\*, C and C++ programming, and distributed database systems development. He graduated with an M.S. degree in computer engineering from Santa Clara University.

**Eric Butler** *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120*. Mr. Butler is an Advisory Software Engineer. He holds B.S. and M.S. degrees in electrical engineering from San Jose State University. His research interests include data center optimization; integrated system, storage, and network management; and storage systems.

**Stefan Jaquet** *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120*. Mr. Jaquet is a Senior Software Engineer. He holds a B.S. degree in mathematics and computer science from Santa Clara University and an M.S. degree in computer science from San Jose State University. He has worked on various data management, storage systems, and storage management projects, and he is currently focused on integrated storage and systems management as well as storage performance management software.

**Madhukar Korupolu** *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120 (madhukar@us.ibm.com)*. Dr. Korupolu is a Research Staff Member. He holds M.S. and Ph.D. degrees in computer science from the University of Texas at Austin, and a B.Tech. degree in computer science from the Indian Institute of Technology, Madras. His areas of interest and contribution are in capacity planning and provisioning (technology released as part of IBM TotalStorage Productivity Center), autonomic resource management and related server and storage optimization in data centers, virtualization management, and more generally, algorithms and distributed systems. He is presently an Associate Editor for the ACM journal *Transactions on Storage*.

**Tapan K. Nayak** *IBM India Research Laboratory, 4 Block C, Institutional Area, Vasant Kunj, New Delhi, 110070 India (tapnayak@in.ibm.com)*. Dr. Nayak is a Research Staff Member. He holds a Ph.D. degree in electrical communication engineering from the Indian Institute of Science, Bangalore. His research interests include resource management, statistical analysis, systems management, and machine learning.

**Ramani Routray** *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120 (routrayr@us.ibm.com)*.

Mr. Routray is an Advisory Software Engineer. He holds an M.S. degree in computer science from Illinois Institute of Technology. His research interests include storage systems, SAN simulation, integrated systems management, machine learning, and disaster recovery.

**Mark Seaman** *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120 (seamanm@us.ibm.com)*. Mr. Seaman is a Software Engineer. He holds a B.S. degree in computer science from Chapman College. He works on solving complex storage management problems.

**Aameek Singh** *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120 (aameek.singh@us.ibm.com)*. Dr. Singh holds a Ph.D. degree in computer science from the Georgia Institute of Technology. His research interests include integrated management and security for enterprise-scale storage and distributed systems.

**Chung-Hao Tan** *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120 (chungtan@us.ibm.com)*. Mr. Tan is a Senior Software Engineer. He holds an M.S. degree in computer science from the University of Southern California. His research interests include HCI, system management, and machine learning.

**Sandeep Uttamchandani** *IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120 (sandeepu@us.ibm.com)*. Dr. Uttamchandani holds M.S. and Ph.D. degrees from University of Illinois Urbana-Champaign. He currently leads the research effort in developing and delivering a resiliency planner for the IBM systems management product line. He has been involved in projects relating to storage protocols, distributed file systems, autonomic storage management, and large-scale customer deployments. He started and developed the SMART project at IBM Almaden Research Center, which explored model-based techniques for storage management. He has authored several papers in key systems conferences and key patent disclosures in the systems management domain.

**Akshat Verma** *IBM India Research Laboratory, 4 Block C, Institutional Area, Vasant Kunj, New Delhi, 110070 India (akshatverma@in.ibm.com)*. Mr. Verma is a Research Staff Member. He holds a B.Tech. degree in computer science from the Indian Institute of Technology, Kharagpur, and an M.S. degree from the Indian Institute of Technology, New Delhi. His research interests include application of algorithmic and optimization techniques to various real-world problems, especially in the area of system management.