Conquer Complexity with Teradata Virtual Storage

Manage Your Hybrid Storage Environment with Ease

By: Ron Yellin
Director, Product Management,
Teradata

TERADATA® | THE BEST DECISION POSSIBLE™

Conquer Complexity with Teradata Virtual Storage

Table of Contents

Executive Overview

Enterprise-class solid state disks (SSDs[1]) have emerged as a new storage technology that will dramatically change how high-performance systems are designed. SSDs are significantly faster than hard disk drives (HDDs), but they are also more expensive. Their adoption will be primarily seen in hybrid (tiered) storage environments (SSDs and HDDs together) as their cost today will preclude their exclusive use for most companies. Figure 1 is an example of a tiered storage environment where the SSDs would be the fastest device, but also the most expensive. The high capacity HDD would be the slowest and least expensive device. The terms hybrid and tiered storage can be used synonymously.

The key to the successful use of SSDs in a hybrid storage environment is having software that will automatically manage data placement and ensure that the most frequently accessed data stays on the faster SSDs.

Teradata Labs developed Teradata Virtual Storage specifically to manage this hybrid storage environment. This paper provides an in-depth overview of Teradata Virtual Storage with insight into its design objectives and a comprehensive view into how it works.

[1] SSDs are a new persistent storage device that uses Flash memory chips instead of spinning magnetic media.

Conquer Complexity with Teradata Virtual Storage

Teradata Virtual Storage is differentiated from other tiered storage solutions by the automation built onto the solution. Managing a hybrid storage environment manually would be an onerous, time consuming task requiring ongoing monitoring and administration. Teradata has created the ultimate ease-of-use hybrid storage management system in Teradata Virtual Storage.

Introduction

The emergence of new storage technologies has provided opportunities for creating new system architectures which can dramatically increase performance, decrease system footprint, and lower costs. SSDs deliver substantially higher performance per device than HDDs but at an increased dollar per terabyte (TB). High capacity HDDs offer lower $/TB but at lower performance per device.

CPU performance has grown at exponential rates over the past six years. The fastest HDDs (15K RPM) have seen only modest performance improvements over the same period. This has resulted in ever larger numbers of HDDs required per processor to support the random I/O performance requirements of enterprise data warehouse (EDW) system processors. At the same time, the disk drive manufacturers have continued to release larger capacity disks and discontinue the lower capacity disks.

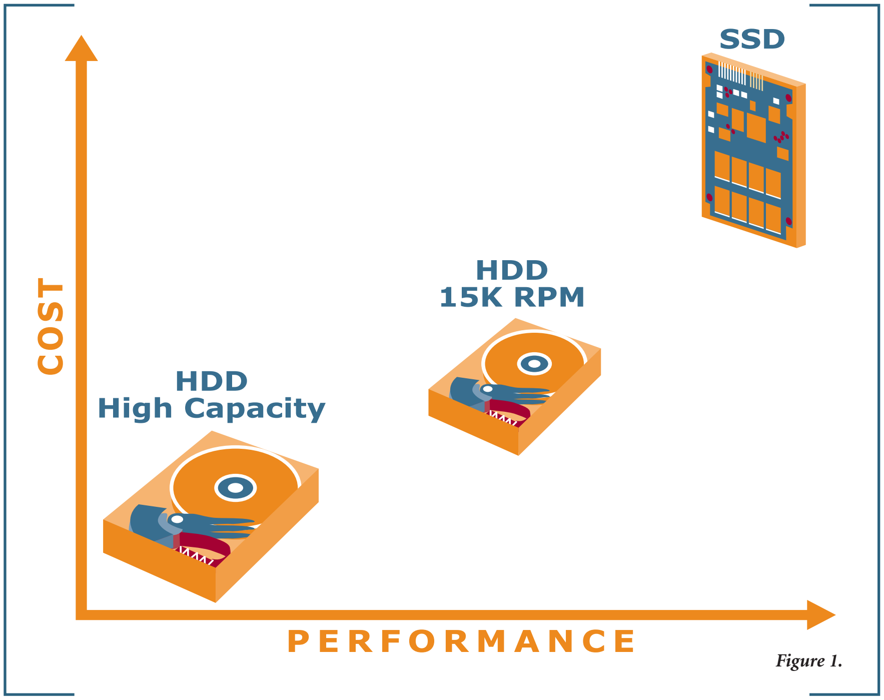

Figure 1.

This trend has caused storage system footprints to grow and CPU per terabyte (a measure of the processing power relative to the amount of storage capacity) to decrease. It also makes the granularity of growth large (adding more CPU power requires more random I/O which drives more storage capacity). While acceptable for many workloads, a growing number of companies do not have the storage capacity requirements to justify increasing storage capacity as their CPU requirements increase.

Each of these new technologies – SSDs and high capacity HDDs – optimizes a key customer purchase requirement (performance or cost), but it is through the combination of these technologies that some interesting opportunities emerge. Mixing the SSDs along with HDDs in a hybrid storage configuration dramatically improves the configuration options. A relatively small number of SSDs can provide all the random I/O performance required to satisfy the demand from powerful CPUs. Combine this with HDDs which can be used to supplement the capacity at a significantly lower $/TB and the result is a system that can be configured with a broad range of CPU to storage capacity ratios. Figure 2 compares SSDs and HDDs in terms of I/O bandwidth and storage capacity.

Hybrid storage solutions can also be made by mixing both fast and slow HDDs into the configuration. The performance characteristics would be less dramatic than the hybrid system using SSDs, but the same concepts apply. Everything discussed throughout this paper with respect to Teradata Virtual Storage would still be applicable.

Conquer Complexity with Teradata Virtual Storage

The key to achieving performance in this hybrid storage environment is having the ability to align the most frequently accessed data with the faster SSDs and allowing the less frequently used data to be stored on the less expensive HDDs.

Teradata Virtual Storage

Recognizing the significance of the emerging storage technologies, Teradata Labs quickly identified that changes to the Teradata Database would be needed to leverage the new storage technologies together in a single environment. Storage in the Teradata Database environment had historically assumed homogeneous devices. To prepare for the future, Teradata Labs initiated a major project to develop a layer of virtualization between the Teradata File System and the underlying storage. This project, which took many years to develop, resulted in the release of Teradata Virtual Storage.

We will discuss Teradata's design philosophy and objectives for Teradata Virtual Storage and hybrid storage systems in the following sections.

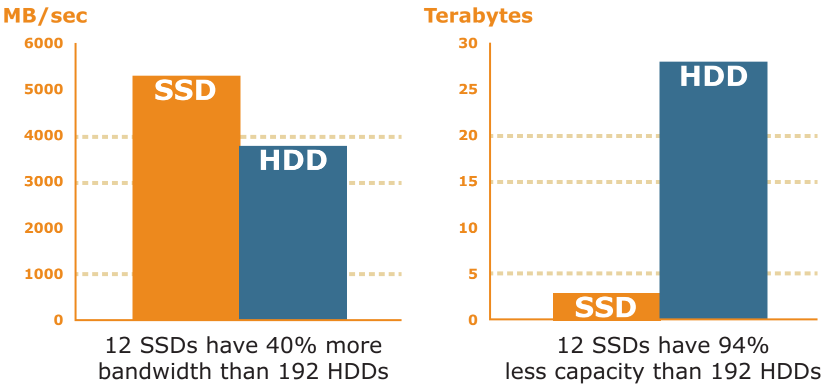

Figure 2. A comparison of SSD/HDD I/O bandwidth and storage capacity.

Automation

As the design process began for Teradata Virtual Storage, automation surfaced as the fundamental requirement for this program. Teradata Labs was not willing to allow Teradata Virtual Storage to result in the creation of a new management task for the DBA. As a result, Teradata Labs would need to design in the ability to automatically keep the most accessed data in the fastest available storage and then move data between storage tiers dynamically as the user access patterns change.

The design would have to ensure that the DBA was not required to be involved in data placement and migration policies. Disk array-based tiered storage solutions used in some other data warehouse environments require the DBA to map their database objects to table spaces and their table spaces to the physical devices. This requires that the DBA understands the access patterns and intended activity level of all objects and ensures that the table spaces are mapped to the appropriate underlying storage to meet the performance and capacity requirements. In addition to ensuring that the storage is sufficient to meet the requirements, the DBA needs to set up the migration policies that govern to which tiers the data can migrate. Disk array-based storage solutions also require that the environment be monitored and managed by the DBA since these environments are susceptible to hot spots (bottlenecks on physical devices) and capacity issues (table spaces running out of space). Furthermore, the DBA would have to continuously adjust the policies because data warehouse workloads evolve and change frequently. User behavior and organizational structure changes can also require policy changes. It's a lot to ask for the DBA to stay on top of the disk array tiered storage management.

For the Teradata tiered storage solution, the objective was to create an environment where no DBA labor would be required. To achieve this, all data placement and migration policies would need to be fully automated. The DBA would not need to define, setup, monitor, or manage the environment.

I/O Distribution Across the Hybrid Storage Devices

There are many factors that influence where the data required for an individual query would reside within a hybrid storage infrastructure. Those factors include total amount of user data, storage capacity in each storage tier, temperature of the data being accessed (how frequently it is accessed), activity level on the system, and the rate at which data access patterns change. With all these dynamic variables, it is not reasonable to assume that all I/Os could always be serviced from the fastest storage tier unless all of the customer's data fit into that tier.

In designing Teradata Virtual Storage and hybrid storage systems, the philosophy was to get the right distribution of I/Os to the right devices. The actual objective was that at least 80% of the I/O workload should be serviced from the fastest tier with the remainder coming from the slower tiers. Knowing that the slower tiers would be servicing some of the I/O workload, hybrid storage environments with both SSDs and HDDs would be configured with enough HDDs to be able to deliver the 20% of the I/O workload potentially not being serviced from the SSDs.

By keeping the most frequently accessed data on the faster SSD, the majority of the I/O workload is offloaded from the slower HDDs and moved to the faster SSDs. This actually results in better and more consistent response times from both the SSDs and the HDDs. As the activity level on the HDDs increase, the queue lengths (pending I/Os for a given disk) increase which lead to slower and more inconsistent response times. This is especially true when a short I/O request associated with a tactical query is last in the queue behind several large decision support I/Os. Moving most of the I/O workload to the SSDs shortens the HDD queues providing better response time with far more consistency.

Configuration Flexibility

One of Teradata's primary objectives in providing hybrid storage solutions was to enable a broader range of configuration options. The prior HDD only configurations had evolved to have these characteristics:

> Growing capacity per node

> Large granularity of growth

> Growing storage footprint

> Decreasing CPU performance per terabyte of storage

The increasing CPU power of the nodes was driving an increase in the number of disks required per node to balance the I/O bandwidth supply with query demands. With more disks per node, the user capacity per node was also increasing. Compounding the capacity growth per node was the introduction of larger disk capacities (and the discontinuation by the disk drive manufacturers of the smaller drives) which further drove up the capacity per node. Figure 3 shows an example of the user capacity per node of HDD-only and hybrid systems. For each HDD disk capacity on the x-axis, the hybrid and HDD-only configurations provide the same I/O bandwidth. As can be seen, the capacity and growth increment of HDD-only systems is large. Using the

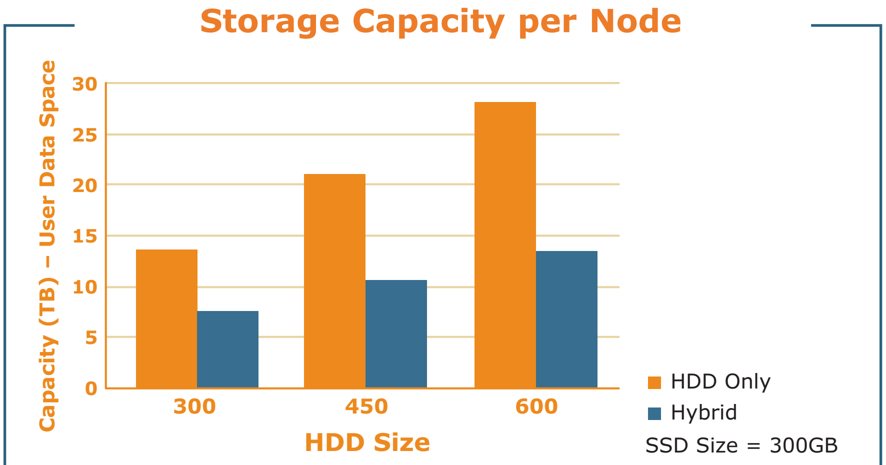

Figure 3. Storage capacity per node.

300GB HDD size, which is the smallest available, each node brings with it nearly 14 terabytes of user capacity.

This capacity growth trend was actually good for some companies that have large storage capacity requirements with relatively light CPU and I/O bandwidth per terabyte requirements. For companies requiring more CPU or I/O bandwidth per terabyte of storage, the only way to achieve this with HDDs is to leave some of the storage capacity per node unused. These configuration requirements can alternatively be met with hybrid configurations that can effectively deliver more CPU and I/O bandwidth per terabyte without the wasted capacity. The lower capacity per node of hybrid configurations can be seen in Figure 3.

The growing capacity per node can also be represented in terms of performance per terabyte of user capacity. Figure 4 compares the relative performance per capacity of the HDD-only system, the HDD-only system with half populated disks, and the hybrid system. In this example, all HDDs and SSDs are 300GB in capacity. As can be seen, companies needing more CPU per terabyte of storage in the HDD-only environment would need to underpopulate the disk drives. By half populating the HDDs, you can realize a two-times improvement in performance per capacity. The hybrid system configuration in Figure 4 delivers greater than a four-times improvement in CPU per terabyte over the HDD-only configuration.

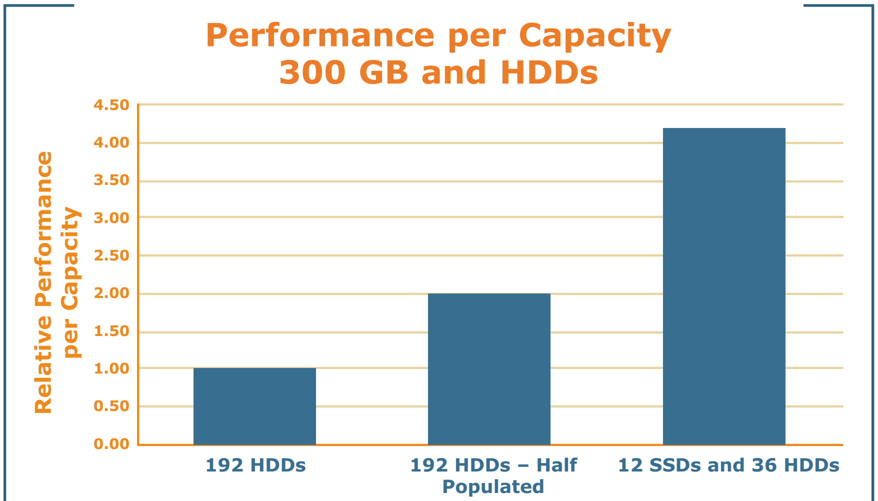

Figure 4. Relative performance per capacity.

As the number of HDDs per node has grown, so has the storage footprint. The storage required for a single server node has grown to consume up to three storage racks. Hybrid configurations can dramatically reduce the floor space and power requirements, which can positively impact the total cost of ownership. Whereas these savings should be interesting to all, it is most compelling for those customers who were driving requirements for higher CPU per terabyte systems that resulted in under-utilized storage capacity.

Building its intelligence into the database software, as opposed to the disk array, provided Teradata Virtual Storage with the added benefit of being independent from the physical disk array. In disk array-based solutions, the movement of data only takes place within a physical disk array. With Teradata Virtual Storage, data can be moved both within as well as between, disk arrays. This flexibility provides the greatest opportunity to optimize data placement within the environment.

Performance

The term performance has many meanings and interpretations. To some, it may mean how much work can be performed on the system while others may view performance in terms of how long individual queries take to execute. This is the classic throughput verses response time discussion.

When comparing homogeneous HDD and hybrid systems, the capacity of the hybrid system needs to be considered. When the number of nodes is equal between these two storage configurations, the capacity of the hybrid system will be less, but the throughput of these systems would be expected to be similar. In this case, however, the hybrid system would offer a smaller storage footprint, reduce the amount of potentially under-utilized storage capacity, and consume less power. I/O-intensive workloads executing against hot data (data which is most frequently accessed) located within the SSDs would be expected to see some response time benefits although the system level throughput would be comparable.

If the homogeneous HDD and hybrid storage systems were configured to have equal capacity, then there would be more nodes in the hybrid configuration since less capacity is configured per node. This configuration would deliver higher throughput and faster response times than the homogeneous HDD system.

Teradata Virtual Storage – A Look under the Covers

Teradata Virtual Storage was designed to integrate tightly with the Teradata Database to deliver a fully automated solution for managing data movement within a hybrid storage platform.

Definitions/Foundation

Next we will discuss the Teradata File System and temperature, which provide the foundation necessary to understand the inner workings of Teradata Virtual Storage.

Teradata File System Basics

The Teradata Database is a shared nothing database that has been built from the ground up to be parallel. Work is evenly distributed across the parallel worker tasks (called AMPs) to dramatically improve query execution time. Each AMP owns a portion of each database object with rows being distributed to the AMPs based on running the row's primary index through a hashing algorithm. The output of the hashing algorithm is the RowID, which contains a row hash that is used in assigning the individual row to an AMP.

Each AMP owns its own physical storage which is only accessed by that individual AMP. Within the storage for each AMP, rows are stored in data blocks that are stored in cylinders and accessed through the Master Index (MI).

A data block contains rows for a single table stored in RowID order[2]. Data blocks are variable sized up to a maximum size of 127.5 KB[3], which is 255 sectors of 512 bytes each. As rows are added and data blocks fill up, they are split into two data blocks each being one half in size. In a mature database, data blocks will average 96KB, which is one half between a full data block and a half data block (after a split).

A cylinder is a unit of storage allocation that is contiguous space on disk. It contains data blocks from one or more tables. Data blocks within the cylinder are always in TableID/RowID order[2]. The cylinder size has traditionally been 3,872 sectors (approximately 1.89MB) but in Teradata 13.10, a large cylinder option was provided to change cylinder size to 23,232 sectors (approximately 11.3MB). Larger cylinders allow an AMP to address more storage, which is required to support the very large disks (i.e., 2TB) which are now available in the market place.

The MI is a list of cylinders for each AMP in TableID/RowID order[2]. Each entry within the MI will contain a cylinder ID (CYLID) plus (among other information) the beginning and ending Table ID/Row ID contained within the data blocks stored in the cylinder.

Within the file system for an individual AMP, a Table is represented as a collection of data blocks in one or more logically sequential cylinders.

When a table uses a Partitioned Primary Index (PPI), rows will be grouped into a partition based on a partition index (i.e., date). Partitions within a table will always remain in order. Within each partition, rows will be stored within data blocks in RowID order.[2] By grouping rows on a partition index, rows within the same partition will be consolidated down to a subset of the data blocks and a subset of the cylinders for that table thus improving access performance.

Figure 5 depicts three cylinders in logically sequential order. Table 1 (T1) is a PPI table with three partitions in partition order. Each of the three partitions has three data blocks. Table 2 (T2) is a non-PPI table with five data blocks.

[2] Discussions about being stored in order implies virtual as opposed to physical order. Order is maintained via the use of pointers.

[3] Maximum data block size is a system parameter, which by default, is set to 127.5 KB which is its maximum setting.

Conquer Complexity with Teradata Virtual Storage

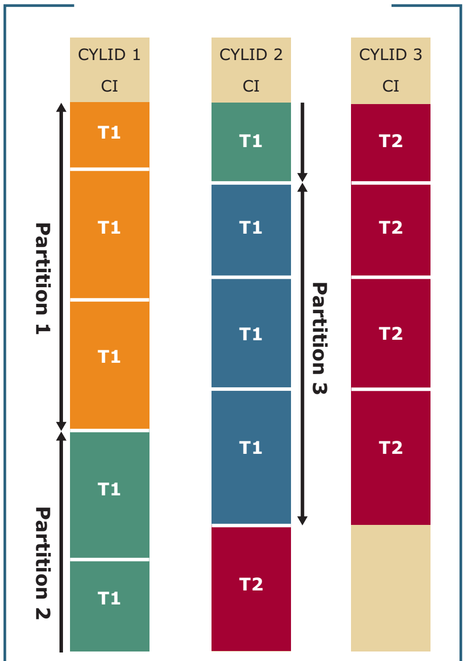

Figure 5. Teradata Database cylinders, tables, partitions, and data blocks.

LUN and Pdisk

A LUN is a logical unit of OS addressable storage typically incorporating multiple devices with raid protection (i.e., a pair of disks in a Raid-1 configuration). A pdisk is a partition of a LUN that is assigned to a Teradata Virtual Storage subpool (see Configuration section for more details). Multiple pdisks can be assigned to an AMP, but each pdisk can only be assigned once. Figure 6 shows a single Raid-1 pair with two pdisks.

Temperature

Temperature is used to represent frequency of access where hot implies the most access and cold implies the least access. Data block access is aggregated up to the cylinder level where a temperature is assigned. Temperature is relative within the system meaning that the cylinders with the most accesses (sum of the data block accesses within the cylinder) will be the hottest and the cylinders with the fewest accesses will be the coldest. Although temperature is managed internally at a very granular level, Teradata Virtual Storage externalizes temperature into three classes: hot, warm, and cold.

Configuration

As was discussed in the Automation section, all aspects of Teradata Virtual Storage have been automated. This includes both initial configuration as well as adding additional storage into the environment.

Teradata Virtual Storage manages storage on a subpool basis. In a multi-clique system, the storage within each clique is considered a subpool. In a single clique system, the storage within that clique is divided into two subpools so that it is possible to cluster across the subpools when Fallback is used. All storage within a system is initially assigned to a subpool.

During the system configuration process, the Parallel Upgrade Tool (PUT) evaluates the storage within each subpool and creates a recommended AMP to pdisk mapping. Teradata Virtual Storage uses this recommendation to determine how the storage is distributed across the AMPs. For each different device type (capacity/performance), Teradata Virtual Storage will determine whether or not the number of pdisks can be distributed equally across the number of AMPs in the subpool. The pdisks that distribute equally are assigned to each AMP's affinity zone. The pdisks that don't distribute equally remain in the subpool and become part of a shared pool of storage that can be used by all AMPs that have access to the subpool.

Today, affinity is set by default to 100%. This means that the pdisks in each AMP's affinity zone are 100% dedicated to that AMP. In the future, Teradata may release systems where affinity is set below 100% which will result in some portion of the storage within each AMP's affinity zone being shared across all of the AMPs within the subpool. A setting of 50% would result in half of the storage being dedicated to the local AMP and half of the storage shared across all of the AMPs in the subpool.

Regardless of the affinity setting on the system, shared storage pools (pdisks that did not distribute across the AMPs evenly) will be treated as 0% affinity which means all of the space will be subject to being allocated to any of the AMPs within the

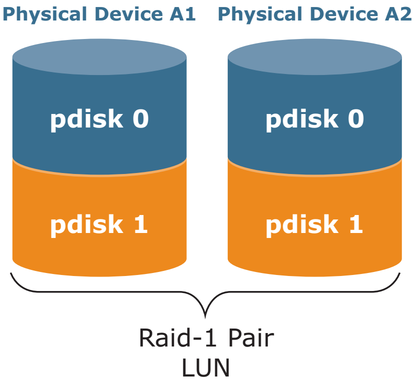

Figure 6. Device/LUN/pdisk.

subpool. Shared storage will be seen as slow by Teradata Virtual Storage so only cold data would be stored in these shared (0% affinity) pools.

Additional storage can be added into an existing clique and will become active upon a system restart. As long as no AMPs are being added or removed, no reconfig is required. Teradata Virtual Storage will move the new storage from the subpool to the respective AMP's affinity zones or retain the storage in the subpool to be used in a shared pool depending on how the pdisks distribute across the AMPs.

As new storage is added into a system, Teradata Virtual Storage will take into consideration existing shared pools of storage in determining what storage goes into the AMP's affinity zone and what remains in the subpool to be used as a shared pool. If the sum of the currently shared pdisks (of one type) plus the newly added pdisks (of the same type) can now be divided equally across the AMPs in the subpool, then Teradata Virtual Storage will automatically move each AMP's pdisks into its affinity zone. The pdisks will now be governed by the affinity setting on the system. If the affinity setting is set to 100%, data from each AMP will begin to migrate back to their pdisks within their respective affinity zones eliminating the shared use of these pdisks.

Device Profiling

As part of the certification process, Teradata Labs conducts extensive performance tests to characterize storage performance based on the array vendor, array generation, physical device type (SSD or HDD), spindle speed, and capacity. The resulting performance data is stored in a repository used by Teradata Virtual Storage during the configuration process to derive a storage grade (speed) for a particular device or location within the device.

During system configuration, the tvsaProfiler will inspect the storage within the clique. Based on the particular storage configuration discovered, the tvsaProfiler will extract the relevant performance data from the repository and translate it into a range of response times for a single device within that configuration. The tvsaProfiler will apply a range of response times to each pdisk taking into account whether it represents a whole physical device or just a portion of a physical device.

For HDDs, the lower Logical Block Addresses (LBAs) of each pdisk correspond to the outermost tracks for that pdisk and are assigned the fastest response times. Figure 7 shows a single HDD LUN (Raid-1 pair) with "N" partitions (pdisks). As indicated in the diagram, the slowest location in pdisk 0 (LBA W) is faster than the fastest location in pdisk 1 (LBA W+1). Each pdisk on a LUN will have different performance levels based on their location within the LUN. At configuration time, the pdisks of different performance levels will be distributed equally across AMPs so they all receive equivalent performance.

Unlike HDDs, SSDs have no moving parts and deliver equal performance to any location within the device. As such, a single response time value is assigned to the entire SSD.

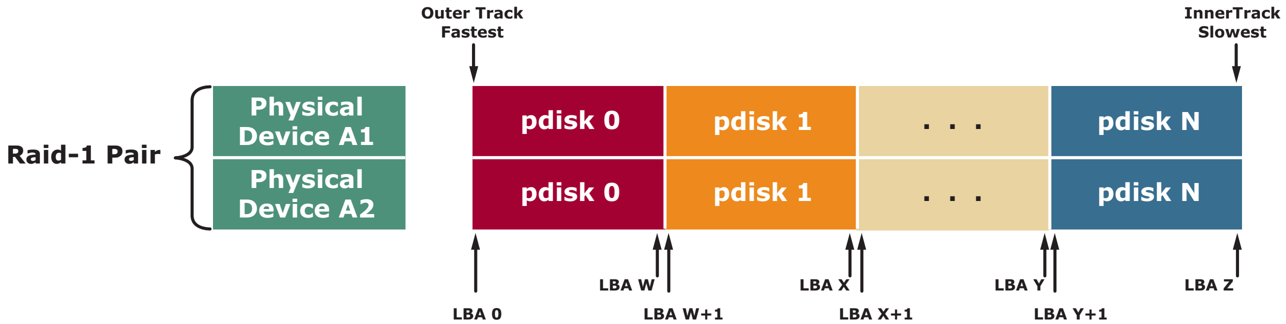

Figure 7. Mapping out the performance regions of a hard disk drive.

Response time values are translated by the Teradata Virtual Storage Allocator and Migrator (described in sections below) into grades which represent relative performance of the storage location.

Soft Reserve and Space Allocation between Storage Grades

Teradata Virtual Storage maintains three storage grades: fast, medium, and slow. Device type (HDD or SSD), rotation speeds (15K, 10K, 7.2K RPM), and physical location on the HDD (outer or inner tracks) all factor into the determination of the storage grade. Within the fast storage grade (SSD), Teradata Virtual Storage maintains a *soft reserve* for the following critical objects:

> **Spool** – Spool tables hold intermediate and final result sets for ongoing queries.

> **WAL (Write after Logging)** – Contains transient journal records needed to roll back open transactions and also contains WriteAheadLog records used to protect in-memory updated data blocks from being lost.

> **DEPOT** – A small area of storage used to protect data blocks which are updated in place. Protects data against disk array errors where existing data can be overwritten but the new data is not written completely or correctly.

These objects are critical to the overall performance of the system, and storing them in the fastest storage has proven beneficial.

The soft reserve, fast, medium, and slow boundaries are determined for each platform and Teradata Database release based on the physical configuration and the relative performance difference between the different grades of storage within the system. Figure 8 shows a sample map for a hybrid system where 25% of the total storage capacity is in SSDs.

When a system contains SSDs, they are considered fast storage. When a shared pool of storage is configured, it will be considered slow storage. The grade for outer tracks of an HDD will be faster than the inner tracks, but the exact grade for HDDs will be determined based on the specific configuration. When there is no shared storage configured into a system, the outer tracks will be medium, and the inner tracks will be slow. When shared storage is configured into the clique, it will be slow, and the inner tracks of the HDD will be medium speed storage.

Temperature Management

Teradata Virtual Storage manages temperature at the cylinder level. There are two mechanisms involved in temperature management called metrics collection (tracks the heating up of cylinders) and metrics aging (manages the decay of temperature as time goes by).

Metrics Collection

Metrics collection is a Teradata Virtual Storage process used to track cylinder accesses. When an I/O is issued from the Teradata Database, the cylinder ID (CYLID) is translated into a device and cylinder number, and a Logical Block Address (LBA) is calculated. The access to that cylinder is logged into an internal memory buffer.

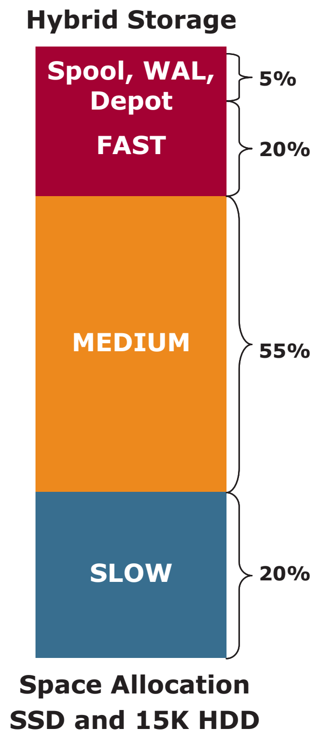

Figure 8.

At least once every minute, the buffer is passed to the Teradata Virtual Storage Migrator module for processing. Temperature management by Teradata Virtual Storage is similar in concept to the stock market's moving average. Based on the length of the time window, Teradata Virtual Storage uses internal algorithms to update the temperature of each cylinder being accessed. As is also true with the stock market moving average, recent activities will carry less weight as the length of the time window increases. Shorter time windows will result in more weight being applied to recent activities.

The time window length is a balance between responsiveness, needless movement of data, and resource consumption. For example, if the time window is too short, a DBA issuing a query against some cold data might result in its temperature increasing enough to trigger the migration of cylinders to SSD. If the access of the cold data was not going to be a sustained activity, it would be better not to consume the resources and move the data in this instance. The window length in Teradata Virtual Storage has been set to react to cylinders that are "heating up" but not to overreact and move cylinders before enough time has transpired to constitute a trend.

Metrics Aging

Using internal algorithms, Teradata Virtual Storage will periodically lower the internal temperature setting of all cylinders in the system. A cylinder's temperature when first assigned is reflective of the moving average discussed in the metrics collection section above. As time goes by, assuming accesses have decreased, the temperature of that cylinder would begin to decay. This decaying of cylinder temperature is implemented by the activation of an aging process which uniformly lowers the temperature of all of the cylinders in the system.

As cylinder temperature gradually decreases, the cylinders being accessed will have their temperatures increased as was described in the metrics collection section. The combination of the metrics collection (heating up of accessed cylinders) and metrics aging (decaying of temperature) maintains a spread in temperatures between cylinders being accessed and those that are not. As was discussed in the definitions and foundation section, temperature is a relative metric. As those cylinders being accessed most frequently maintain the highest temperatures, Teradata Virtual Storage will be able to ensure they remain aligned to the fastest storage locations.

Migration

Maintaining cylinder temperature is only one part of the work done by the Migrator. As its name suggests, the Migrator uses the cylinder temperature to decide which cylinders need to be migrated to different speed storage. For each AMP, the Migrator maintains an ordered queue of misplaced cylinders. The hottest, or most frequently accessed, cylinders that are stored in the slowest storage locations are placed first in the queue. This primary migration function does not target cold data located on fast storage. Although these cylinders could also be considered misplaced, moving a cylinder which is not being accessed to slower storage has no immediate benefit. The one instance where a cold cylinder located on fast storage would be targeted is when there is no room in fast storage to move in a hot cylinder from slow storage. In this case, the migration becomes a two-step operation where the cold cylinder is moved to slow storage, and then the hot cylinder is moved into the vacated location in the fast storage.

The Migrator and Allocator (see the Allocation section for more details) functions are part of the TVS VPROC[4] process which runs on each Teradata Database node[5]. The Migrator will queue up one migration operation per AMP every five minutes. A migration operation is defined as either a single step migration where a hot cylinder is moved to faster storage or a two-step migration where a cold cylinder has to first be migrated out of the way. Each Migrator will only perform two concurrent migrations. The Migrator will continue processing its queue of required migration operations until all targeted cylinders have been migrated.

At this default migration rate, approximately 10% of the allocated cylinders on the system can be migrated in a week. System overhead to perform metric collection, metric aging plus migration at the default rate is about 2% of the CPU and I/O. The benefit realized by aligning the cylinders to the correct storage will typically far outweigh this cost.

Teradata Virtual Storage supports a faster mode of migration called *optimize*. In optimize mode, the Migrator will use all available resources to migrate data as quickly as possible. Approximately 10%

[4] The TVS VPROC will exist in all Teradata 13 and beyond systems regardless of whether Teradata Virtual Storage is activated. When Teradata Virtual Storage is not in use, the device profiling, temperature management, and migration functions are not active, but the underlying infrastructure required to support the Teradata Virtual Storage functionality is being used.

[5] Depending on the actual configuration, a Teradata Database node will have either one or two TVS VPROC processes.

of the allocated space on the system can be migrated in about eight hours. This mode will impact system performance so it should not be used while production workloads are being executed. This operation is sometimes called burst mode.

The Migrator will suspend migration when the storage capacity of any AMP reaches 95%. Although this suspension is Migrator local, AMPs across the system tend to fill up at the same time so migration will effectively be suspended system wide.

Database writes to a cylinder being migrated will be held until the cylinder has been migrated and then they will be applied. New migrations to a cylinder will be held until in-progress data base writes complete.

Up until this point, the migration discussion has pertained to the regularly scheduled (i.e., one migration operation per AMP each five minutes) migration operation. In addition to this migration mode, there is an asynchronous migration mode that operates at a rate greater than the regularly scheduled migration and is initiated when capacity in soft reserve, fast, or medium storage reaches 90% full. The purpose of this asynchronous migration mode is to free up space in faster storage for new allocations of hotter data. This migration mode will continue until all target zones (soft reserve, fast, or medium) have 10% free space or until there are no more cylinders subject to migration. Hot data will not be migrated out of soft reserve and fast zones, and warm data will not be moved out of medium storage.

Allocation

Allocation is the process of allocating storage capacity to the AMPs. When an AMP requires additional storage capacity due to the currently accessed cylinder being full, a request is initiated from the Teradata File System for a new cylinder to be allocated. The allocation request will contain a temperature parameter indicating the desired temperature which is based on the intended use of the new cylinder.

The Teradata Virtual Storage Allocator module receives the request and, in turn, with help from the Migrator, will allocate the AMP a cylinder whose performance characteristics (fast, medium, or slow) match the requested temperature if available. If the targeted storage grade has reached capacity and is not available, the next slower available storage grade will be allocated. The cylinder will maintain the targeted temperature designation even though it is stored on slower storage. This misplaced cylinder will become eligible for migration.

Temperature Defaults

The desired temperature issued by the Teradata File System is based upon defaults established for the platform. As can be seen in Figure 9, the default temperature for Perm Data is dependent on the physical configuration. When the number of hard disk drives per node divided by the number of AMPs per node is less than two, the default perm data temperature is hot (# HDDs / # AMPs < 2). When the

Default Allocation Temperatures

	Hybrid System SSD and HDD
Perm Data	**If #HDDs/AMP >= 2; WARM** **If #HDDs/AMP < 2; HOT**
Spool	**Hot**
WAL	**Hot**
Depot	**Hot**
Global Temp	**Warm**
Permanent Journal	**Hot**

Figure 9. Default allocation temperatures.

number of hard disk drives divided by the number of AMPs is larger than or equal to two, the default perm data temperature is warm (#HDDs / #AMPs >= 2). As an example, take a system that has 30 HDDs per node and is configured with 30 AMPs per node. In this environment, the default perm data temperature would be hot since 30 HDDs divided by 30 AMPs equals one which is less than two.

Initial Data Temperature

When data is loaded into a new table, Teradata Virtual Storage does not initially know anything about the data's intended use. Data will initially be loaded into storage for the default Perm temperature as specified in the table above. Once loaded, Teradata Virtual Storage will determine the data's appropriate temperature by monitoring its accesses and then move the data to the appropriate storage over time.

A feature of Teradata Virtual Storage called Initial Data Temperature (IDT) provides a means for allowing the DBA to override the default Perm temperature and to specify an initial temperature. This would typically only be used when the DBA knows that the data being loaded is cold and should be stored on slow storage. In this instance, it can be more efficient to load directly into the slow storage instead of first loading into the Perm data default temperature and then having to migrate the data out over time. The IDT feature can also be used to make newly loaded tables for a new application for future use warm so as to not crowd out current hot data.

Whereas IDT provides the DBA some measure of control, the philosophy and value proposition of Teradata Virtual Storage is in its automation. The controls provided by IDT should only be used in very specific use cases where the data temperature is known and it differs from the default Perm data temperature for the platform.

IDT is implemented via the "TVSTemperature" query band and will override the default Perm data temperature setting for data loading into new tables and will remain in effect for the life of the session or transaction depending on its use.

When there is any doubt about the initial temperature, system resources will be better utilized if the data is loaded into storage for the default Perm temperature and migrated as appropriate.

A second component of the IDT feature of Teradata Virtual Storage is the FORCE command within the Ferret Utility. This command can be used to change the temperature definition of a table, partition (for a PPI table), of range of row IDs (for a NoPI table). This command can be used to change the temperature of existing data as opposed to the TVSTemperature Query Band which effects data temperature at load time.

When the FORCE command is used, only the temperature designation is changed but the data is not actually moved. By changing the temperature designation, the affected cylinders may become eligible for migration (assuming the new temperature designation does not match the current storage location) by the standard Teradata Virtual Storage migration feature.

When designating a temperature with IDT using either the TVSTemperature Query Band or the FORCE command, it only sets the initial temperature. The data is then subject to Teradata Virtual Storage's Temperature Management function which will dynamically change the cylinder over time based on access frequency.

As was the case with the TVSTemperature query band, the use of the FORCE command should only be considered for unique cases where the automated migration is not meeting some specific requirements. There may be some desire to use the FORCE command to pre-heat some cold data (for year-over-year or month-over-month analyses) in advance of its intended use but overall system throughput will be better served if Teradata Virtual Storage is allowed to manage temperature automatically and only migrate data based on its use. Use of the FORCE command may cause thrashing by causing cylinders to migrate in and out of SSD independent from their actual usage.

Teradata hybrid systems are designed with the assumption that a portion of the required I/O bandwidth (approximately 20%) will be supplied from the HDDs. A common misconception is that acceptable performance will only be achieved if all accessed data comes from the SSDs. This is clearly not true. As was discussed in the I/O Distribution Across the Hybrid Storage Devices section, the HDDs in a hybrid configuration are more than capable of delivering acceptable performance as the majority of the I/O workload is offloaded from the HDDs and moved to the SSDs. This leaves shorter queue lengths on the HDDs which allows for more consistent response times.

Loading Data into the Teradata Database

When data is loaded into the Teradata Database, its initial temperature will depend on whether the load is going into a new table or whether data is being added to an existing table. In addition, it will depend on whether the TVSTemperature query band was issued. What happens in each of these situations is discussed in the sections below.

Loading into New Tables

When a new table is defined, a table header will be written into the last allocated Perm data cylinder for each AMP (if space exists). If you recall from the File System Basics section, tables are maintained in order within the cylinders so the newly defined table will begin in the last allocated cylinder after the last data block from the previously defined table.

As data is loaded into data blocks following the table header, the existing allocated cylinder will run out of space, and a new cylinder will be allocated. If the DBA had issued the TVSTemperature query band, the new cylinder will be allocated with the specified temperature. If no query band had been issued, the default Perm data temperature will be used.

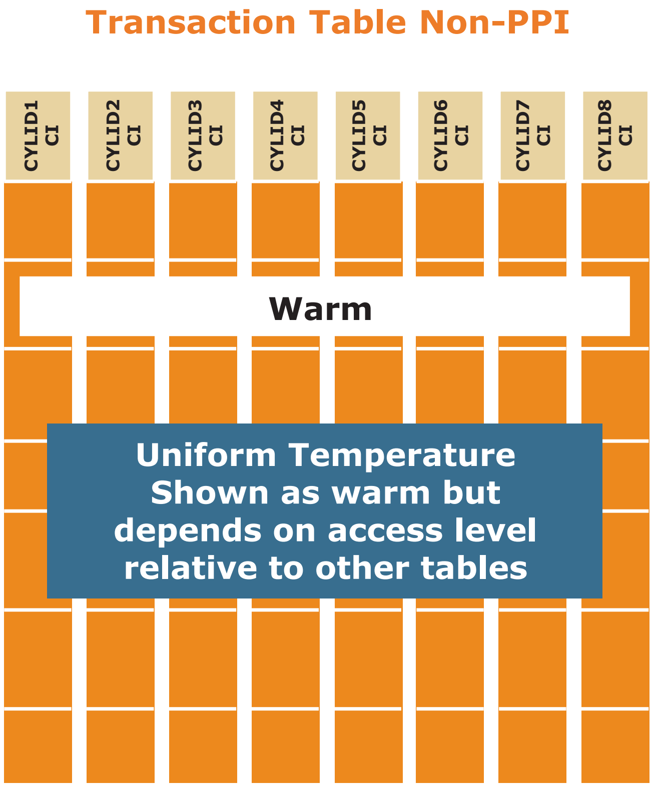
Transaction Table Non-PPI
CYLID1 CI
CYLID2 CI
CYLID3 CI
CYLID4 CI
CYLID5 CI
CYLID6 CI
CYLID7 CI
CYLID8 CI
Warm
Uniform Temperature Shown as warm but depends on access level relative to other tables

Figure 10.

If the first cylinder used for the new table was pre-existing when the table was created, it will already have a temperature and will not use either the query band or the default Perm data temperature setting. All subsequent allocations for that table will follow the query band (if present) or default temperature as indicated above.

Loading into an existing table containing no rows will be treated as a new table and either honor the TVSTemperature query band if present or use the default Perm data temperature. A previously defined empty table may be followed in the cylinder by another table. In this case, the table header and data blocks for the next table will be moved into a new cylinder making room for the new data blocks to be added into the existing table. The new cylinder will be kept in logically sequential order within the Master Index.

If during a load, the storage grade for the targeted temperature reaches capacity, the loads will be directed to the next slower available storage grade. The cylinders will maintain the targeted temperature designation even though they are stored on slower storage. These misplaced cylinders will become eligible for migration.

Loading into Existing Tables

Loading additional data into an existing table differs from the new table load discussed in the previous section. Existing tables already have allocated cylinders with defined temperatures. The new rows are merged into the existing data blocks and cylinders and kept in order based on their TableID and RowID. Their temperature is that of the existing cylinder in which the rows are stored. The TVSTemperature query band, if issued, is ignored when loading into an existing table.

When an AMP needs more storage for that table, a new cylinder will be allocated. It will retain its logical order within the Master Index. The new cylinder will inherit the temperature from the original cylinder (the cylinder that reached capacity and resulted in a new cylinder having to be allocated).

Tables without PPI will tend to have uniform temperatures across the table since queries against a table tend to touch a large percentage of the cylinders. This occurs since the rows are stored in row hash order within the data blocks and cylinders which is random with respect to any query. That is, all queries accessing a subset of rows will do so in a random pattern against the entire table. A non-PPI table can be seen in Figure 10. In this example, the uniform temperature for all of the cylinders making up this table is warm. The actual temperature will depend on the access frequency relative to other cylinders within the system.

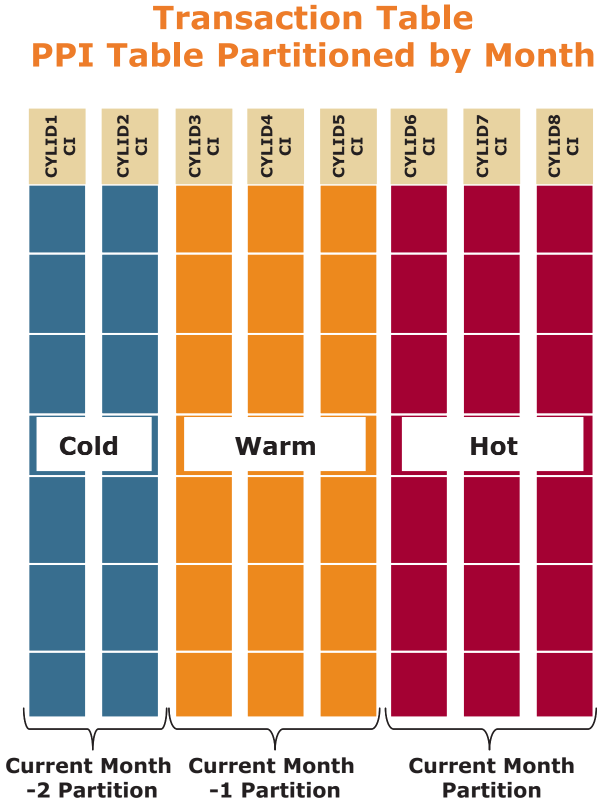

Figure 11.

Small tables without PPI have a greater chance of being hot and thereby being stored on SSD since the random accesses across the cylinders would be going against a smaller number of cylinders. For very small tables, the Teradata Database optimizer will choose to scan the whole table thereby heating up all of the cylinders comprising that table. A large table would require very heavy access before the random access pattern would cause all of the cylinders to heat up enough to be considered hot.

Larger tables without PPI would often be accessed through a secondary index. When a secondary index is used, accesses to the base table are reduced and instead, accesses are concentrated on the secondary index. As a result, the secondary index cylinders will likely be hot and thereby stored on the SSD.

When a PPI table with a partitioned index, which generates increasing partition numbers (like date / time based values), is used, loads will tend to be consolidated to a single partition based on the partition index (i.e., date). Existing partitions will already be defined in existing cylinders and will have defined temperatures.

When loading into a new partition, the partition will be established directly following the previous partition within an already existing cylinder. The next table header and data blocks which might have existed in that cylinder are moved to a new cylinder. The temperature of this new cylinder, as well as any additional cylinders needed as the data in this latest partition grows, will inherit the temperature from the current cylinder.

When date is used for the partition index, loads will go into the latest partition which will generally be hot (most current data). New cylinders for that partition will be allocated as hot since they inherit the temperature from the previous cylinder for that partition. The TVSTemperature query band is ignored when loading into an existing PPI table.

If the PPI partition index is date/time based, and the queries primarily access the most recent data, the cylinders across the PPI table will not have uniform temperatures. Since the Teradata optimizer will eliminate partitions not required to satisfy a query, the older partitions will see decreased accesses which will result in the cylinders comprising the older partitions cooling down. The more current partitions will tend to be the hottest, which will be aligned with the fastest storage by Teradata Virtual Storage. Figure 11 shows a PPI table partitioned by month. In this example, new data will be loaded into the current month. Assuming that accesses begin to decrease as the partitions age, the previous months of current month -1 and current month -2 have cooled from hot to warm and cold respectively.

Conclusion

Teradata Virtual Storage conquers complexity and allows the complete automation of the management of hybrid storage environments providing superior ease of use. By design, there are virtually no knobs for the DBA to turn. Teradata Lab's philosophy is that hybrid systems should be self managed. When manual controls are provided, the tendency will be that the DBA feels compelled to monitor and manage the environment.

The purpose of this paper is to provide insight into the inner workings of Teradata Virtual Storage so that DBAs can grow comfortable with Teradata Virtual Storage's automated approach and thereby understand why manual management of the environment is not required.

About the Author

Ron Yellin is the Product Management Director for the Teradata Platform Infrastructure group within Teradata Labs' product management organization. Ron and his team of product managers are responsible for all Teradata storage and backup and recovery solutions. In addition to leading the group, Ron has individual responsibility for the product management of the Teradata Virtual Storage product. Ron joined the company in 1980.