Realtime publishers

The Shortcut Guide to Balancing Storage Costs and Performance with Hybrid Storage

sponsored by

Dan Sullivan

Copyright Statement

© 2014 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers its web site sponsors. In no event shall Realtime Publishers or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

Chapter 3: Hybrid Storage for Database Servers

Databases are critical components of many enterprise applications. For decades, database administrators have struggled to tune database management systems in order to meet business requirements for performance and consistency. Complex software, such as database management systems, is difficult to tune because so many factors contribute to performance. Database systems make heavy use of persistent storage systems for application data as well as supporting data, such as indexes and transaction logs. As a result, the overall performance of a database server is strongly shaped by the performance of the storage system.

This third chapter focuses on the storage systems for database servers. In particular, the chapter addresses:

- Business requirements of database servers
- Performance impact of disk-only storage systems on database performance
- Benefits of hybrid storage for database performance

Hybrid storage systems can help businesses achieve their database performance requirements without incurring excessive costs or significantly disrupting current operations.

Business Requirements of Database Servers

Database servers are rarely used in isolation in production business environments. Typically, databases function as part of a multi-tiered application that includes one or more user interfaces, Web servers, application servers, directory servers, message queues, and other application support services. These components function together to implement business processes, such as selling products, managing inventory, and recording financial transactions.

Just as workflows across an organization depend on different departments completing their tasks in a consistent, reliable, and predictable manner, so to do enterprise software applications require components that deliver consistent, reliable, and predictable performance. Figure 3.1 illustrates this idea.

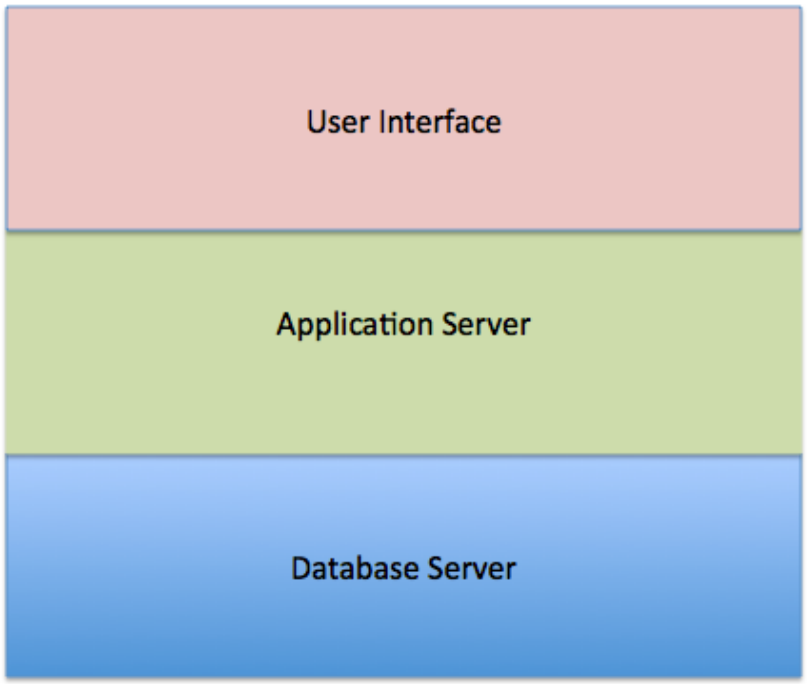

Figure 3.1: Database servers function as one component in a multi-tiered environment.

Variation in the performance of the database server can cause applications to wait longer for data retrieval at certain times. This variation can lead to inconsistent response times to the user interface. User interface designers may have set limits on the amount of time that a user should wait for a response from the interface. If the database cannot perform consistently, the application server may be slowed and that, in turn, delays updates to the user interface. As a database is commonly at the bottom of the application stack, its performance problems can have ripple effects through all other tiers.

For database servers to meet their general requirements, they must be able to deliver:

- Consistent IOPs
- Support for a mix of read and write transactions
- Predictable performance

Each of these factors contributes to the overall performance of the database, which in turn, influences the overall performance of the application.

Consistent IOPs

Consistent performance when reading from and writing to storage is an important factor in database performance tuning. When a database server can deliver a consistent level of IOPs over extended periods of time, then database administrators can tune other aspects of the database.

For example, when a storage system consistently provides 50,000 IOPs, the database administrator can estimate tuning parameters for other components, such as the length of the queue for pending read and write operations or the optimal size of data blocks that are read from and written to the storage device.

When IOPs are not consistent, overall database performance can vary. The time required to perform a read operation from a disk, for example, can vary. If the data block to be read is near the read/write heads of the disk, the time to perform the read may be relatively short. If the data block to be read is farther away, it will take longer to perform the operation. This situation can lead to variability in application response times, as mentioned earlier.

Support a Mix of Read and Write Operations

Database applications, especially those designed for transaction processing, must support a varied mix of both read and write operations. The proportion of reads and writes will vary by application, but it is common to have more reads than writes. For example, a typical workload for a transaction processing application may consist of 70% read operations and 30% write operations.

Because of the way relational databases are designed, a single logical write operation can lead to multiple physical write operations. The physical write operations include writing the raw data to a data block, possibly updating indexes on disk, and writing data to archive logs that are used to ensure data integrity.

Predictability

The third major business requirement of database servers is predictability. This requirement applies to both read and write operations under a variety of circumstances, including:

- Queries that generate large result sets
- Bulk data load operations
- Large numbers of end user connections

Queries that generate large result sets will perform a large number of read operations. Bulk data loads can produce significant numbers of write operations. End user connections can put a mix of both read and write operations on the database server. Although the number and mix of operations will vary over time, performance should be consistent.

The overall goal of the three business requirements (consistent IOPs, support for a mix of read and write operations, and predictability) is to have predictable and reliable business operations implemented in software. When database performance is unpredictable and inconsistent, it is difficult to maintain a well-managed business process.

Having outlined key business requirements, it is time to shift focus to consider how the disk storage system can affect the overall performance of these business operations, and more specifically, how they can affect database server performance.

Performance Impact of Disk-Only Storage Systems on Databases

Storage systems have an impact on many functions of databases. Users may find themselves waiting for extended periods of time for queries to finish because many data blocks must be read to respond to the query. Database administrators might find that backups are taking longer than the time allotted for the operation because disk performance is insufficient to keep up with growing data volumes. Developers tuning queries may find that slow-running queries are not caused by poorly crafted SQL code but by long queues of disk I/O operations waiting for processing. Getting useful estimates of insert and update times may be difficult because the performance of writing to transaction logs is unpredictable.

The performance impact of disk-only storage systems on databases falls into three categories:

- Read and write performance
- Lack of predictable performance
- Tuning challenges

Each of these performance factors can be addressed to some degree but the fundamental nature of disk drives imposes limits on the overall performance of disk-only storage systems in databases.

Read and Write Performance

The performance of read and write operations is highly influenced by the physical design of disk-only storage systems. To appreciate the performance limits of disk drives, it helps to review the design of disk-based storage systems. Similarly, the physical characteristics of flash devices and hybrid storage systems offer significant performance advantages over disk-only storage systems.

Basic Disk Drive Components

Disk drives are composed of:

- Platters
- Spindle
- Spindle motor
- Read/write heads
- Actuator

Drives vary in the materials used and other design characteristics, but the overall structure is fundamentally the same (see Figure 3.2). It is the physical limitations of the components and how they operate that lead to the performance limitations experienced by database users.

Platters are circular devices coated with a magnetic material that stores the actual data on a disk. Platters are distinguished by the type of material used (e.g., aluminum, glass, and ceramic), the thickness of the platter, and the platter's heat absorption characteristics.

A set of platters is kept in place by a spindle. The spindle motor rotates all platters on a spindle in unison. The speed of the spindle motor determines the rate of rotation of the platters, such as 5400RPMs and 7200RPMs.

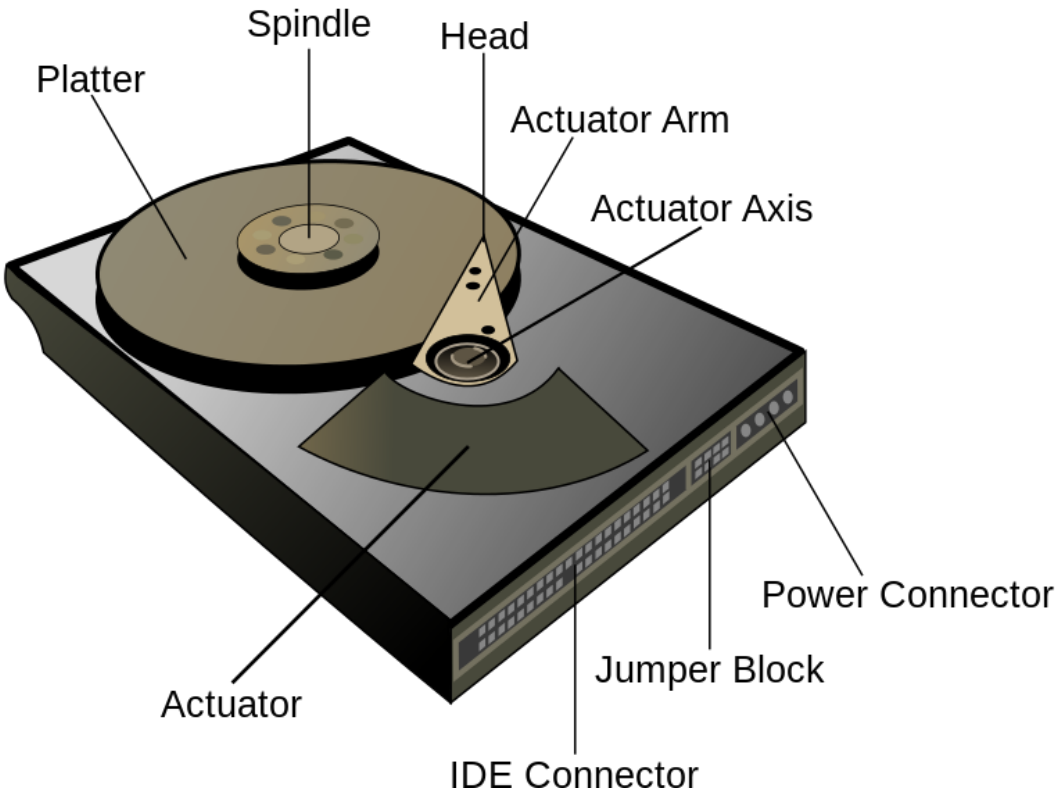

Figure 3.2: Basic components of a basic disk drive (Source: http://en.wikipedia.org/wiki/File:Hard_drive-en.svg).

The read/write heads sense and change the magnetic fields on platters in the process of reading and writing from a disk. Each platter typically has two read/write heads, one for each side of the platter. The read/write heads are all attached to a single actuator axis so that the heads move together and are always in the same relative position on each platter. The actuator positions the read/write heads over specific areas of the disk as needed to read from or write to that position.

Performance Impact of Physical Characteristics of Disk Drive

The performance of disk drives is highly dependent on physical properties of the components. The time required to write data to a position on the disk is determined, in part, by the magnetic materials used as well as construction of the read/write heads. More important, read and write performance is limited by the rotational speed of the platter and the time required for the actuator to move the read/write heads.

Another factor that limits the performance of disk drives is the fact that a single actuator axis is used. The read/write heads cannot move independently; therefore, while the read/write heads might move closer to a target data block on one platter, the heads might move farther away from another needed data block on another platter.

Read/Write Scenarios

To further highlight the importance of the physical characteristics of disk drives on performance, consider the following three scenarios:

- Multiple applications reading from a disk
- A single query reading a large volume of data
- An application writing multiple data blocks

These scenarios represent data access patterns that can occur with disk-only storage systems.

Multiple Applications Reading form a Disk

Multiple applications can be reading and writing to a database server at any time. Analysts might be running ad hoc queries, database administrators are loading data, and background processes of the database server are performing routine operations, such as writing transaction log data to disk. This example scenario considers only read operations. Figure 3.3 shows a simple illustration of how data blocks may be distributed over a disk.

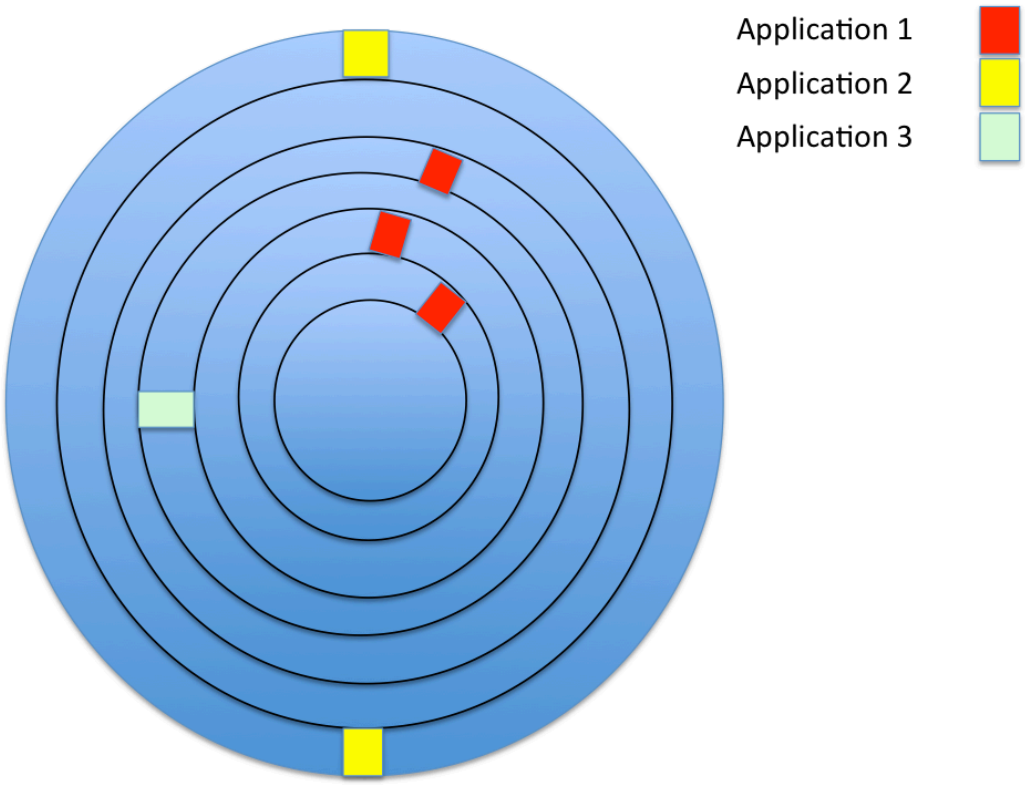

Figure 3.3: Data blocks associated with different applications are distributed across the disk.

As the application generates read and write requests, they are queued to be performed. The database may start with Application 1, depicted in Figure 3.3, and read the innermost red bloc. It may then need to read the blocks of Application 2 on the outermost edge of the platter. Reading the other two blocks of Application 1 follows this operation. Finally, the disk repositions the read/write head to read the one data block associated with Application 3.

This pattern of read operations requires multiple movements of the read/write heads. First, the heads have to move toward the center of the platter to read the first block. The read heads have to move toward the outer edge of the platter and wait for the platter to rotate until the next data block is in position for reading. After reading one of the data blocks associated with Application 2, the disk has to wait until the platter rotates 180 degrees before it can read the second data block.

This pattern of access leads to multiple movements of the read/write heads and time spent waiting for the platter to rotate to the next position. Waiting for these movements contributes to the latency of reading from the disk. Not all access patterns share these back-and-forth, waiting-for-the-platter-to-position characteristics.

A Single Query Reading a Large Volume of Data

Business intelligence and data warehouse queries often require a large volume of data. For example, an analyst might want to compare year-to-year sales growth in a particular line of products. Data warehouses that aggregate sales data by individual product for each day for each sales outlet will have many database records to retrieve.

A common way to populate a data warehouse is with bulk data loads. When writing data to the drive, the database may be able to allocate contiguous blocks of data. When this occurs, the distribution of data is more localized to a part of the disk, as Figure 3.4 depicts.

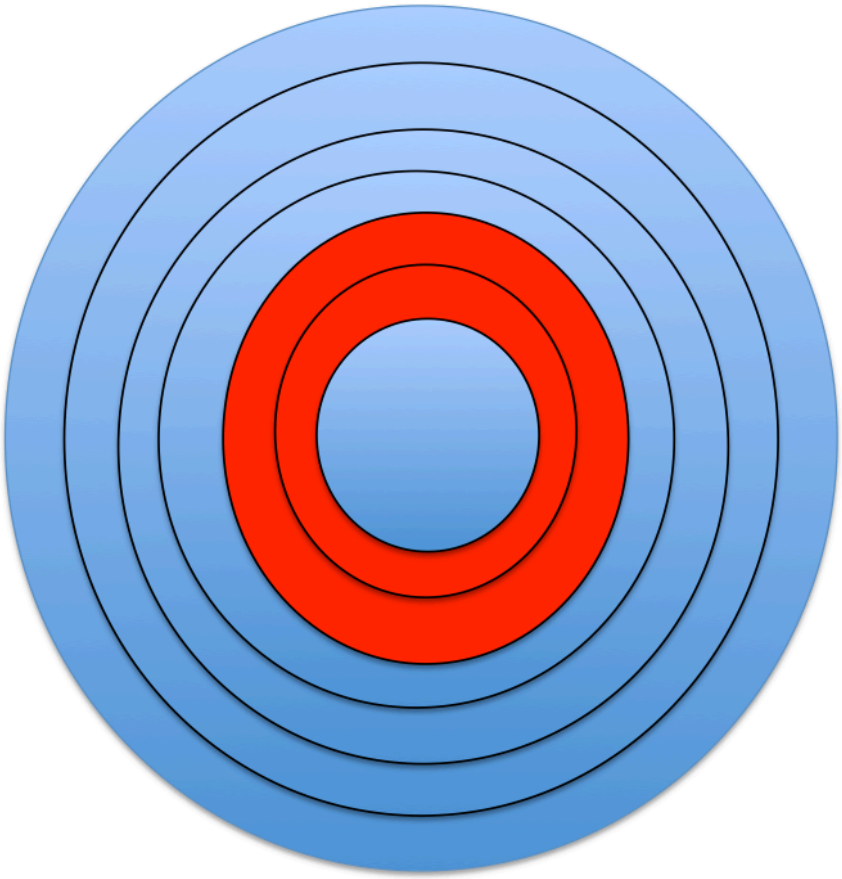

Figure 3.4: Contiguous data blocks are adjacent to each other and can be read or written with minimal movements of the read/write heads.

Reading contiguous data blocks reduces the need to move read/write heads to different parts of the disk. Once the read/write heads are in position, the data can be read as the platter rotates. There is no need to wait for the read/write heads to move after reading each data block. In addition, the data blocks are read as the platter rotates, so there is no time spent rotating without reading a data block.

This situation depicts and optimal scenario. Not all database applications use queries that read large blocks of data. Writing large amounts of data in contiguous blocks is not always an option in some database applications. Also, even if data is originally written in contiguous blocks, changes to the data can lead to fragmentation of data blocks. The performance of write operations can also suffer because of the distribution of data already written to disk.

An Application Writing Multiple Data Blocks

Reading and writing share some of the same performance characteristics because read/write heads must move into position prior to performing the operation.

When database processes write to disk, there must be a sufficient number of free data blocks to write the data. If substantial amounts of data have already been written to the disk and some data has been deleted, there is likely to be fragmentation on the disk.

Fragmentation occurs when data blocks are deleted, freeing up the data block. Data blocks adjacent to the freed block may or may not be free. When free data blocks are sparsely distributed over the disk, it can take longer to write data to the disk. The reason is that after writing a data block, the disk must wait for read/write heads to reposition and the platter to rotate to the correct position before performing the write operation.

Figure 3.5 shows an extreme case in which most of the disk is full and few data blocks are free. If all the free blocks were written to, the read/write heads would have to move from the outer edge to the inner track of the platter. In addition, the write operations would have to wait for several full rotations to be in position to write to each of the free blocks.

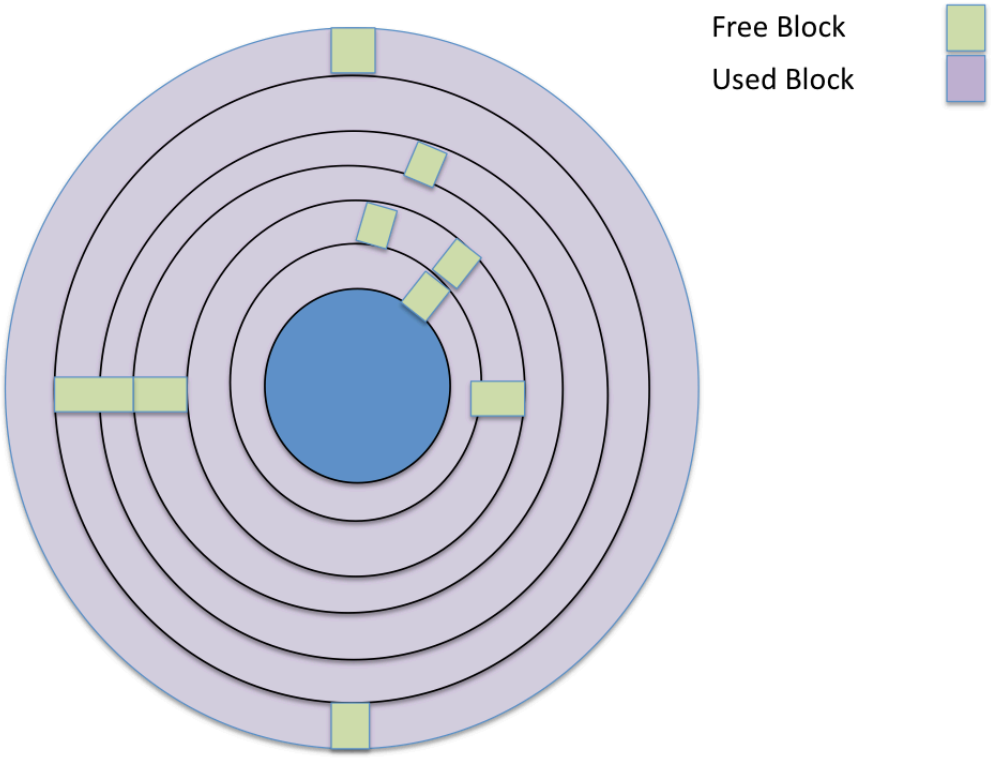

Figure 3.5: Writing to disk requires finding free or available blocks that may be distributed across the platter.

As these scenarios show, the physical characteristics of disk-only storage systems can lead to different performance outcomes. Performance depends on the placement of data on the disk and the read and write patterns generated by database applications. These dependencies, in turn, lead to two challenges: lack of predictability and difficulties in database tuning.

Lack of Predictability

Businesses are best able to manage business operations when they are predictable. Predictability allows you to build complex workflows with sufficient confidence that components within the workflow will function as expected. The need for predictability and related attributes, such as reliability and consistency, drive the use of service level agreements (SLAs).

IT professionals often work with SLAs because they provide a means of defining levels of functionality and performance that satisfy the business needs while offering measurable characteristics that IT professionals can use to design, build, and manage applications and infrastructure. The lack of predictability in disk-only storage systems makes it difficult to achieve the consistent, predictable performance needed to meet SLAs.

Tuning Challenges

The physical limitations of disk-only storage and the lack of predictability in performance present a number of tuning challenges. Database administrators typically seek to maintain performance targets for both transaction processing systems and analytic applications, such as business intelligence systems. There are limits to how much a database administrator can do.

Some enterprise-scale databases have self-tuning features that help administrators determine key parameters, such as data block size and memory allocation. Administrators can also make decisions that minimize contention for disk resources. For example, data and indexes may be stored on different devices to reduce the chance of contention when both the data and indexes must be read in the same operation.

Developers can invest significant amounts of time in tuning queries. Often, the goal of tuning is to reduce the number of I/O operations performed by the query. For example, if fewer than 10% of the rows of a large table are returned by a query, performance can be improved by implementing an index. This setup can result in the need to read fewer data blocks and a reduction in the latency of the query.

Of course, it is important to use all compute and storage resource efficiently, but database tuning does not have to be a moving target. Lack of predictability and issues with read and write performance can be avoided with hybrid storage systems.

Benefits of Hybrid Storage Systems

Hybrid storage systems combine flash and disk technologies to provide improved performance benefits over disk-only storage. The use of hybrid storage systems is particularly useful for data-intensive applications such as database management systems. Several benefits stand out in particular:

- Improved I/O performance
- Ability to scale up IOPs as needed
- Cost benefits of disk storage for large data stores
- Performance benefits of flash without excessive costs

These benefits stem from the physical characteristics of flash devices as well as the architecture of hybrid storage systems.

Consistent I/O Performance

As noted in the discussion about disk-only storage, consistent I/O performance is an important characteristic of database servers. Flash devices use a fundamentally different design than do disk systems; flash device design avoids the performance-varying features of disk systems.

Flash devices store data in transistors by applying an electrical charge to gates in the transistor. Reading from and writing to these gates does not require any physical movement of components, as is the case with disk drives. This lack of movement eliminates the variability in the read and write operations that can occur in disk-only storage.

Data is retrieved from disks using protocols that impose additional overhead on read and write operations. Retrieving data from PCIe flash devices does not have that same overhead. In fact, server-based flash devices can use fast PCIe channels to move data from flash storage to a host CPU, as depicted in Figure 3.6.

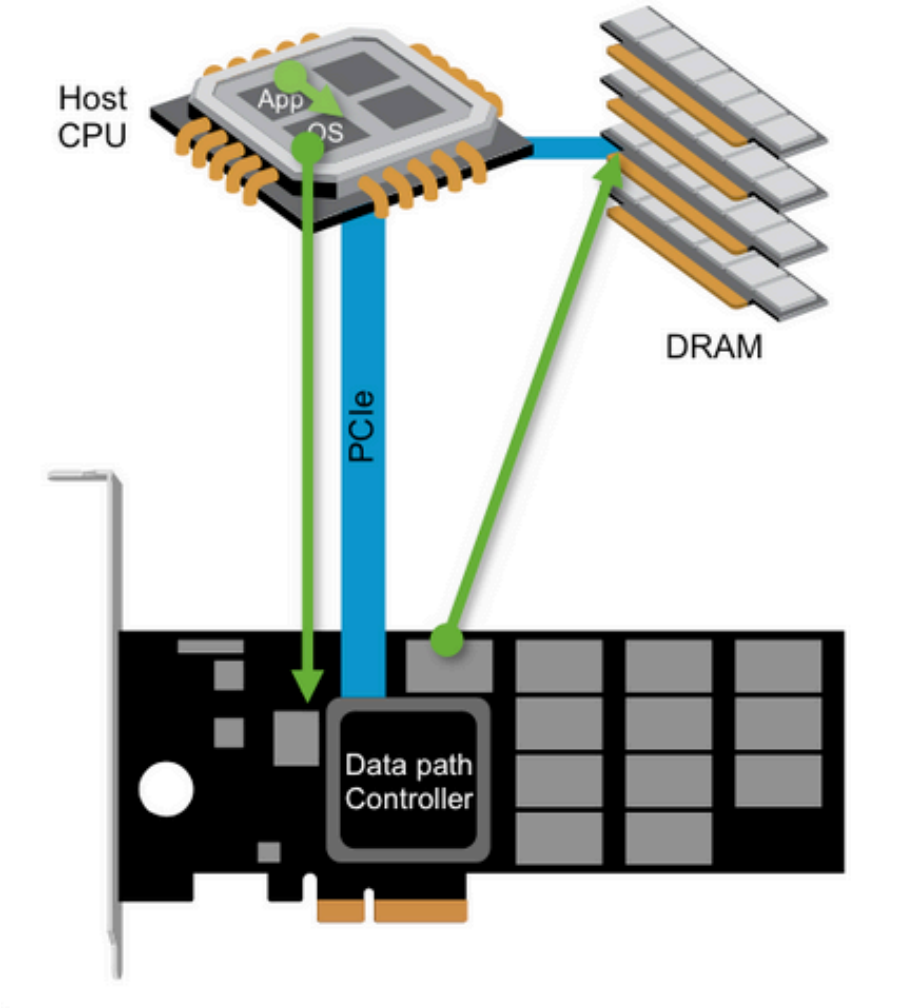

Figure 3.6: Server-based flash storage can transfer data to and from the CPU using the PCIe at faster speeds than a comparable operation on a disk-based system.

With the right design choices, a hybrid flash-disk storage system can also provide consistent I/O performance. The key is to isolate disk operations from database operations. Rather than write directly to a disk, databases can write to the flash device, which provides for fast and consistent write operations. After the write-to-flash operation is complete, the data can be transferred to disk. The performance issues associated with disks still exist but they do not affect the database. The flash device in a hybrid system decouples database performance from disk performance.

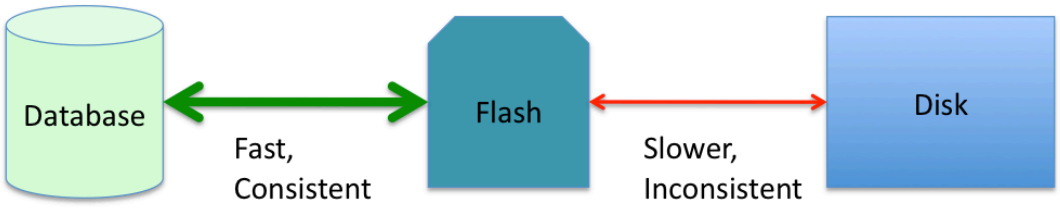

Figure 3.7: Hybrid storage systems decouple inconsistent disk performance from database applications.

Flash also improves read operations. Data that has been read from disk is cached in flash storage. When the data is needed again, as is often the case in database applications, it is read from the flash device rather than from the disk. Although the initial read of data may be from the disk, subsequent reads will complete with lower latency.

Ability to Scale Up IOPs as Needed

Flash devices can provide substantially more IOPs than can disk-based storage systems. When a hybrid storage system is designed with software to allocate IOPs to different processes, the IOPs dedicated to the database can be scaled up as needed.

For example, a data warehouse may start with a modest amount of data and a small number of users. In such cases, a single flash PCIe device can meet the performance requirements of the project. As the volume of data and number of users grows, additional flash PCIe devices can be added to provide linear scalability in performance.

Flash storage complements random access memory with regards to caching. Database management systems are designed to cache hot data in RAM, but the total available memory may be insufficient for the amount of hot data in use. Flash storage on the server can perform a similar type of caching, supplementing the database-managed caching.

Cost Benefit of Disk Storage for Large Data Stores

A distinguishing feature of hybrid storage is that it preserves much of the cost benefit of disk storage relative to more expensive flash devices. Hybrid storage systems employ disk systems for large volume, persistent storage. As more storage is required, additional disks can be added to storage arrays at lower cost than adding flash devices. This, however, does not mean there will be a performance penalty.

Consider a transaction processing system writing a new record to the database. Regardless of the size of the data stored on disk, the new data will be written to the flash device first and the database will treat the data as committed to the database. As long as there is sufficient space on flash storage, the database will experience consistent storage performance.

Similarly, when data is read from disk, it is cached on a flash device. In a data warehouse, the most frequently accessed data may be the latest data. This reality is not surprising because managers and analysts will want access to the most recent information about business operations. When database users repeatedly seek the same data, it is considered "hot." Hot data is typically a small fraction of the total amount of data stored in a database. Flash devices on a database server will cache this hot data and allow faster access to the data than if it were retrieved from disk.

Performance Benefits of Flash Without Excessive Costs

Hybrid systems provide the performance benefits of flash devices without incurring the cost of a flash-only storage system of comparable size. The reason for the benefits of flash without the cost has to do with the architecture of a hybrid system.

Flash devices in database servers act almost as an extension of memory from a caching perspective but have the persistence characteristics of disk storage. A well-designed hybrid storage system uses flash storage for caching hot data and performing write operations so that databases do not experience long latency during I/O operations. By decoupling database performance from disk performance, hybrid systems provide consistent, low latency IOPs as well as the lower cost storage capacity of disk systems.

Summary

Business applications require consistent IOPs, support for a mix of read and write operations, and predictable performance. Disk-only storage systems are limited in their ability to deliver consistent performance by their physical characteristics. The mechanical parts of disk systems function at much slower speeds than do the electronics of flash devices that have no moving parts.

Although an all-flash storage system may be an ideal, the cost can be prohibitive. Well-designed hybrid systems offer the benefits of flash devices without the excessive costs. A key factor driving this cost-benefit combination is the efficient use of flash for storing hot data and enabling fast write operations. Data is moved to and from disk as needed but often in ways that do not keep database applications waiting. Hybrid storage systems can help improve the overall performance of database servers without excessive cost.