How To Configure Oracle FS1-2 Storage Based On Data Skew; the Most Important New Metric In Array Configurations
# Introduction

## Common Data Access Patterns in IT Environments

- **OLTP Transactional Data**
- **Business Intelligence/Data Warehouse Data**
- **Financial Services**
- **Telco Processing**
- **Virtual Desktop Infrastructure**
- **SPC-1 Benchmark**

## Lessons Learned from Access Patterns

## Data Skew in Depth

- **Skew Definition**
- **Skew Tipping Point (Diminishing Return Point)**

## Flash Capacity Calculation: The Roll Down From Highest Tier to Lowest Tier

- **Sizing Each Tier**
- **Calculating the Write Penalty**
  - **Formulae**
  - **Write Penalty**
  - **Back-End Calculations**
  - **Common Sense Modifications**

## Cooling Periods

- **Cooling Curves**
- **Oracle Database Cooling Considerations**
Introduction

In the past it was easy to figure out how many drives of a certain type were needed to deliver a specified number of IOPS. Since a data volume existed on only one type of media, storage administrators could divide the number of IOPS a hard disk drive (HDD) was capable of delivering into the total number of IOPS needed and instantly know the minimum number of drives required. This, plus a small compensation for various read/write ratios and RAID types, and the calculation was complete.

With the advent of auto-tiering arrays, however, this heretofore simple problem becomes much more complex. Now storage decision-makers and administrators need to ask: “How much of up to four different types of media are needed to deliver \( nn.nnn \) host IOPS, and what is the optimal combination of these differing media types.” Additionally, since these tiers all can have different RAID types, the RAID write penalty has to be applied to specific data on specific media. And, hot data (accessed frequently) is always changing, so it’s always moving.

The critical question is, "How much of the data is hot, how much do I have and where should it go?" Behind this question is the fact that not all data is accessed equally. Some is very active data, some is less active, and some is stone cold. This is not a static model. Blocks of data are always changing—some cooling, some heating, and some remaining constant. The challenge is to know which blocks of data are which and to place them on the optimal media to meet an organization's performance and business requirements in the most cost-effective, efficient manner. Like many things, this is easy to say and hard to do.

The difference in data access density, or frequency of use, within a specific set of data blocks, is called skew. A simple definition of skew is that it describes the following situation: a small amount of the data associated with any given application typically accounts for the vast majority of I/O’s issued against the total amount of capacity. The higher the variance, the higher the skew.

Nearly all workloads exhibit significant data skew. Given the huge differences in performance and capacity costs for storage media, understanding an application's data skew is critical. Why would organizations ever place all of their data on one type of media? To do so means typically paying far too much for capacity while being forced to provision expensive, high-performance media to store cold data that requires low I/O performance.

A storage strategy that does not take into account the dramatic I/O skew in data inevitably leads to unnecessarily expensive over-provisioning and a probable inability to handle unexpected or
unplanned peak workloads. Either or both of these outcomes can lead to reduced end-user satisfaction and possibly truncated career paths.

This paper is written for storage administrators and IT executives who want to understand data skew and how it will influence your storage strategy. The paper examines several common data access patterns in several industries, explores the concept of a "skew tipping point," presents methods of sizing each tier, and provides insights into data cooling rates and data growth rates. Additionally, this paper introduces the concept of sub-LUN auto-tiering combined with the Quality of Service Plus (QoS Plus) feature, which is implemented in the Oracle FS1-2 flash storage system. QoS Plus is specifically designed to dynamically manage data placement based on its skew and business value.
Common Data Access Patterns in IT Environments

The following examples of data access patterns demonstrate the importance of data skew, and how it affects performance and capacity costs. A more detailed explanation of skew can be found in the section of this paper called "Data Skew in Depth."

OLTP Transactional Data

In OLTP environments, data is hot during the transaction, warm during the billing month (shipping, restocking, store dailies, etc.), and cold (read only) until monthly/quarterly/annual reports are run. After this, data is stone cold—99.99 percent of transactional data is never updated. Exceptions are reversed or corrected transactions, and these are rare. Data access is random during transaction, and then access tends to be sequential as data ages after the billing cycle, which may be, in some cases, cotemporaneous with the transaction.

The overall skew for pure OLTP transactional data is high. Many of these transactions are closed and paid for immediately via Foursquare or PayPal or via a credit or debit card, and they're out of the hands of the selling merchant and into the hands of the financial services industry. The average data cooling period is approximately 30 days, so throughout a calendar year one-twelfth of the data is potentially hot at any given time—only 8.3 percent of all the data! Actually, this percentage actually may overstate the amount of hot data since the average latency time from transaction to monthly billing period is really only 15 days given a reasonable, equal distribution of atomic transactions over any given 30-day period. 4.15 percent is probably a better measure of hot data. This ratio assumes that the total capacity is based on everybody who transacted at least once during a calendar year. The actual capacity is larger as there are dormant accounts on the books, which will be weeded out eventually by the merchant. This paper does include bias for different weed-out periods.

Business Intelligence/Data Warehouse Data

Typically this data is created via transactional activity and now is in a data warehouse where most accesses are sequential in nature for reporting purposes or analytics. Data cools as it ages and becomes less germane. After a year it's also stone cold. The overall skew is medium to low. The longer the data is kept in the warehouse and accessed, the lower the skew. This is the realm where Oracle Exadata and Oracle Exalogic excel. Oracle Exadata for its extremely high internal bandwidth and high degree of parallelism, and Oracle Exalogic for its ability to analyze large amounts of data quickly and efficiently.

This use case is not well exploited by auto-tiering unless the array is also serving as a disk archive for cold transactional data.

Financial Services

The vast majority of financial services transactions (excluding high-frequency trading that is an extreme corner case) are demand deposit accounting (DDA) transactions in the form of checking, savings, ATM services, and credit/debit card processing, an electronic form of DDA.

The data is hot while a transaction is in process, and again later that day when it is posted and the bank closes its daily cash positions and updates all retail balances. Once a month there is a flurry of activity as statements, either paper or electronic, are prepared and monthly reporting is done. Quarterly and annual reporting operates off an extracted subset of the raw data that is typically stored on different volumes and used by different applications, so the reports create more hot spots, albeit in different places.
Card balances must be billed and paid, so a billing cycle heats up one-tenth of the data every three days, typically based on the least significant digit or the checksum digit on a credit or debit card. The other nine-tenths of the data is just forlornly biding its time waiting to be rescued from a life of pitiful insignificance buried on a disk drive in a data center somewhere in the world. Payments are processed as they arrive, so this is random, but generally one-thirtieth of the data is accessed every day based on a 30-day accounting month. After these steps, activity devolves to read-only reporting and analytics queries.

The overall skew is high. Data is accessed multiple times in the first 30-45 days so it stays quite warm, then it turns cold (15 days on average to next billing period, and 15 to 30 days to monthly reporting). This is considered an extended cooling period. The significance is that 45 days is one-ninth of a year, and if a year represents 100 percent of the data at any given time, one-ninth of all data is reasonably active and about half of that data is very active.

Telco Processing

The core Telco call detail records (CDRs) processing application is one of the most demanding in the world. Calls are placed and CDRs are collected and stored by the switches. Once a day, the CDRs are transferred to the main processing system and remain there until the billing remediation process (BRP) starts. There are 10 BRP runs per month. Each is three days in length and is based on the least significant digit in the bill-to phones number. So one-tenth of the monthly data is accessed during each three-day cycle. The other nine-tenths of the data are dormant, awaiting a moment of glory.

A flurry of activity takes place when call tariffs are applied. Payments to foreign carriers are calculated, wherein complex international wireless calls might involve 10 carriers in five countries. This alone generates huge amounts of new data in addition to the preparation of the bills to the phone subscribers. Bills are sent and eventually paid, which causes the last major flurry of mainline activity. Like financial services, the activity is fairly even across the month, so one-thirtieth of the data heats every day. CDRs and BRPs are archived after this and extractions to go into data warehouses for later analysis and reporting, or they are delivered to certain large federal agencies for other purposes.

The overall skew is high. The data cooling period, similar to financial services, is somewhat extended—to around 45 to 60 days. Activity ratios similar to financial services apply here.

Virtual Desktop Infrastructure

A virtual desktop infrastructure (VDI) is the classic case for which all-flash arrays (AFAs) are said to be the sine qua non of solutions. But this requires examination. The VDI problem has always been "boot storms," wherein nearly all employees show up at work at about the same time and fire up their workstations. Suddenly hundreds of users are trying to access the same golden image from the same set of disks at the same time. The 512-byte reads used in booting cause a massive disk overload and traffic jam. If a single laptop can boot Windows up in two minutes, 100 users on the same set of disks can boot up in what...100 minutes or 50 minutes? Whatever the length of time, it's far too long.

The overall skew is very high, perhaps as high as 99 percent for a large VDI installation.

SPC-1 Benchmark

Ninety-six percent of SPC-1 IO's access only 5.5 percent of the data—a benchmark that is no longer meaningful in an auto-tiering world. Here's why.

No more farcical configurations based on hundreds of 146 GB HDDs. Instead, a configuration based on a few lightning fast 400 GB performance flash drives in the Oracle FS1-2 and everything else on 4 TB NL_SAS drives. The 4 TB NL_SAS drives cost $.25/GB, 30X lower than high performance flash and they are there just to show a
large capacity and really low $/GB; the 400 GB performance flash drives deliver the high IOPS bragging rights. SPC-1, RIP 2014.

Lessons Learned from Access Patterns

Several conclusions can be drawn from the access pattern examples.

In the case of VDI, an alternative solution is to place the boot images on inexpensive read-optimized capacity flash. Approximately one-half of a TB, or 500 GB, ought to do it for 1,000 users. This covers the golden boot image and all shared applications, including Microsoft Exchange, SharePoint, and DLL libraries.

With respect to user data in VDI environments, consider the sustained I/O rate per user to "My Documents" or "My Scans" or "My Pictures." Generally accepted statistics say ~4 IOPS/second per user for heavy users is probably realistic. **Four IOPS/second per user!** Flash is not the recommended solution. Also, consider shared data directories, for which capacity must be balanced with sufficient performance so no one's productivity is compromised. Users may type a paragraph in Word, fix spelling, type a complex formula into Excel (five embedded IF statements and debugging), and then hit the save icon to prompt disk I/O. Actual, sustained user IOPS are very low. No reason to buy an all-flash array. Instead, a flash LUN of ~500 GB can be allocated on Oracle FS1-2, at one-fifth the cost or less.

The important take-away from the access pattern examples is that truly hot data seems to be within a range of 4 to 6 percent, with perhaps 80 to 90 percent of all IO’s falling into this range. Historically the change rate of customer data, as measured by delta change replication techniques shows a consistent rate of between 3 to 5 percent of all data included within the protected volume set. This posits that 4 to 6 percent new hot data may be accurate.

Multi-tiered file systems and archive managers (in Oracle's case, this is StorageTek Storage Archive Manager) have shown consistently for more than 20 years a 3 to 5 percent ratio of new, hot, changing data to archive-class data. What this demonstrates is that truly hot data, which needs SSD-like performance, comprises a very small percentage of the total data set. It also confirms that all-flash arrays are probably a bad idea because organizations pay flash prices for the 95 percent of data that does not need flash performance. Said differently: "A little flash goes a long way."

The data access pattern examples should be a wake-up call. Expensive flash storage can be managed wisely, and this is precisely what the Oracle FS1-2 flash storage system does—with no hidden charges and no software capacity penalty. It is the first array on the market that offers both performance and business intelligence.

Data Skew in Depth

Skew can be high, or skew can be low, so storage decision-makers need to ask themselves, "What is the optimal amount of flash storage to be configured?" and "What metrics do I need and how do I obtain them?" Before these questions can be answered, it is necessary to define the metrics of skew, then to consider how they are calculated and how to apply the law of diminishing returns to the addition of flash. The goal is to configure enough flash to get exceptional benefit without over-configuring flash to hold tepid or cold data.

Skew Definition

There is no recognized definition of skew that is commonly used to describe nonlinear disk access across any given amount of capacity. The higher the skew, the more non-uniform the disk accesses tend to be. A different and much more useful term can express skew as a numerical value. The term we will use is: skew tipping point, and it is defined as follows: the percentage of IO’s plus the percentage of total capacity to which those IO’s access is equal to 1. An example: 85 percent of IOPS are contained within 15 percent of the capacity, resulting in .85 + .15 = 1. The
skew tipping point is 85 percent\(^1\). This is a simple but useful way to look at skew since if either of the values is known, users can back into the other value and easily figure out how much tier-zero SSD storage is needed.

Currently there are no tools that explicitly calculate the skew tipping point or determine exactly what the skew is for any given application or set of applications. The previous section of this paper provided reasonable skew assumptions based on common workloads. Storage decision-makers can discuss how the data flows through the application(s) to verify assumptions. If the assumptions are within the boundaries discussed earlier—3 to 6 percent hot data—they are probably OK. If they are far outside of these boundaries, the assumptions should be re-evaluated prior to finalizing configuration designs.

Skew Tipping Point (Diminishing Return Point)

Figure 1 illustrates skew tipping point (Diminishing Return Point). The lower left corner represents 0 percent IOPS at 0 percent capacity, while the upper right represents 100 percent of the IOPS at 100 percent of the capacity. The addition of the two figures of merit (.85 and .15 in this case) results in a range of zero to two. The left hand part of the curve (green arrow) represents the area where adding a little bit of SSD capacity has a high impact on performance (high utility). As one goes to the right (blue arrow), the utility of additional SSDs decreases as there are very few IOPS accessing the data. The Diminishing returns point is where the sum of the percentage of capacity and the percentage of IOPS is equal to one. This is not necessarily the perfect point, but it serves to deliver a very good approximation of where to stop adding SSD capacity.

![Image of skew tipping point](image)

Figure 1. Illustration of skew tipping point

Now it’s abundantly clear that the majority of data is cold and consumes few IOPS. To return to an important question, “Does it make sense to place all data on an all-flash array (AFA) when most of the data is not hot?” No. Placing cold or lukewarm data on expensive flash media is a waste of money that could be better allocated elsewhere in an enterprise.

\(^1\) While the Skew Tipping Point (Diminishing Return Point) is expressed as a very specific value, in real life, it represents a range of values that will cluster around this locus; sometimes a little more, sometimes a little less. It should be taken as a reasonable starting point in calculating how much flash storage will deliver high value to your business.
Many AFA vendors tout integrated deduplication and compression as the "secret sauce" that makes their products more affordable. Achieving 5x data reduction of changing hot OLTP data is probably generous, but so be it. To add insult to injury, the performance impact of constantly deduplicating and rehydrating active data is not zero, and this affects the "data sheet performance" they are trying to sell.

A 5x data reduction, by definition, reduces storage costs by 5x. But the cost differential between 4 TB capacity HDDs and flash memory is 30x. Given that in most cases 95 percent of the data is cold and does not require flash performance, why would anyone settle for a 5X reduction in cost for 100 percent of data compared to a 30x reduction in cost for 95 percent of the data. The math is obvious.

In this skew tipping point example, if one wants to capture 85 percent of the IOPS generated on flash, one would provision sufficient flash to equal 15 percent of the total capacity. In real life, as discussed earlier, the diminishing return point tends to be far more aggressive. The vast majority of observed I/O skew is represented by 3 to 6 percent of the data. Diminishing return points of ~95% (SPC-1 benchmark) are not uncommon. A little flash goes a long way!

Flash Capacity Calculation:
The Roll Down From Highest Tier to Lowest Tier

The basic way to estimate what capacity is needed for both flash and multiple lower tiers is to calculate the amount of storage needed for each tier by looking at the amount of total IO’s that needs to be absorbed by each tier (tier absorption rate). For example: If an environment has a skew tipping point of 80 percent with a total required amount of IOPS equal to 50,000, the top tier has to absorb about 80 percent of the IO’s or 40,000. The remaining lower performance tiers have to cover the remaining 20 percent or 10,000. When there are multiple tiers, users should assign 80 percent of the remaining 20 percent, or 16 percent, of the IO’s to the next tier and repeat this process until the lowest tier is reached.

Sizing Each Tier

To size each tier, it’s useful to understand the performance difference between flash and HDDs. The numbers in Table 1 represent recently observed testing and are an average of multiple workloads. They are not datasheet numbers, which do not represent real-world application performance; rather, they represent single-drive laboratory performance in a test jig measured at the drive interface, not at the application interface. Block sizes are in the 4 to 8 K range. RAID write penalties vary by RAID type and are discussed below.

<table>
<thead>
<tr>
<th>Media Type (per drive)</th>
<th>IOPS²</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance SSD</td>
<td>15,000</td>
<td>70/30 Read Write</td>
</tr>
<tr>
<td>Capacity SSD</td>
<td>9,000</td>
<td>90% Read</td>
</tr>
<tr>
<td>Performance HDDs</td>
<td>225–250</td>
<td>70/30 Read Write</td>
</tr>
<tr>
<td>Capacity HDDs</td>
<td>90–100</td>
<td>70/30 Read Write</td>
</tr>
</tbody>
</table>

² The IOP values shown above are representative of values that can be achieved across a variety of workloads for SAN attached storage measured at the host. Flash data sheets often show higher values, but they represent a single, empty drive being tested in a custom laboratory jig and will not be realized in real application situations.
Read/write ratios are critical in calculating the actual number of "back-end" IOPS as compared to "host IOPS" since various RAID types incur read-modify-write overheads (RAID write penalties) as described below. These need to be taken into consideration.

<table>
<thead>
<tr>
<th>RAID Type</th>
<th>RAID IO’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 10</td>
<td>2</td>
</tr>
<tr>
<td>RAID 5</td>
<td>4</td>
</tr>
<tr>
<td>RAID 6</td>
<td>6</td>
</tr>
</tbody>
</table>

Most OLTP workloads are 80/20 or 70/30 read/write. For certain workloads, like serving files over an NFS network, the back-end read/write ratio is ~50/50, which incurs a large RAID write penalty. Additional disk IOPS are needed to deliver the host-facing NFS operations required. Other applications that produce higher write-to-read ratios are certain VDI implementations and many mail servers.

Remember that all types of flash are fantastic readers, and while performance type flash drives are also good writers, Capacity flash drives are typically slower, but they are improving all the time.

Calculating the Write Penalty

This simplified example demonstrates how to calculate the needed IOPS and capacity per tier for any given application. The assumption is a skew of 80 percent and a read/write ratio of 80/20 percent. The formulae show how to calculate the amount of flash needed inside the flash pool as well as the amount of other storage needed in the lower tiers.

Formulae

\[ 0.80 \times 50000 = 40,000 \text{ IOPS} \]

is what is needed inside the highest tier of the flash tier due to the 80 percent skew factor. For the remaining 10,000 IOPS, the following rule is applied: Base the remaining data placement of 80 percent of the remaining IOPS on the next lower tier and repeat this until the lowest tier is reached.

The next tier has 10,000 IOPS remaining to absorb:

\[ (0.8 \times 10000) = 8,000 \text{ IOPS for performance SAS} \]
\[ (0.2 \times 10000) = 2,000 \text{ IOPS for capacity SAS} \]

Without the write penalty applied, the tiers are as follows:

- Flash: 40,000 IOPS
- Performance SAS: 8,000 IOPS
- Capacity SAS: 2,000 IOPS
- Total: 50,000 IOPS

Write Penalty

The above calculation looks good! The 50,000 IOPS requirements are spread across all three tiers, but this is not representative of what the array has to do because of the RAID write penalty. To help storage decision-makers understand the importance of this factor, the effective write IOPS of a 4 TB capacity HDD must be considered. The assumption is that read IOPS are 90\(^3\) per second, so with RAID 6, the write IOPS are an amazingly low 15—a 6x penalty. The back-end load on the disks is higher than the host-driven I/O’s, often by a surprisingly large amount.

3 Some people say 100, some say 110, or even 120. This depends on seek distance and the number of commands queued in the drive. This mode is conservative, as mentioned before. It is always better to under promise and over perform versus the alternative.
Back-end load means that the write penalty is included in calculating the exact number of spindles needed. Because this example includes approximately 20 percent writes there is a need for more back-end IOPS than the 50,000 IOPS coming from the host application.

Back-End Calculations
The back-end calculations are as follows:

SSD tier:

\[(0.8 \times 40,000) + (2 \times 0.2 \times 40,000) = 32,000 + 16,000 = 48,000\] IOPS for the SSD tier (400 GB SSD in this case)

which is RAID 10 in this example, or

\[(0.8 \times 40,000) + (4 \times 0.2 \times 40,000) = 32,000 + 32,000 = 64,000\] IOPS using RAID 5. (This is significantly higher than the 40,000 host IOPS calculated initially.)

HDD tiers:

The following RAID 6 example shows 2.5” 10 K HDDs:

\[(0.8 \times 8,000) + (6 \times 0.2 \times 8,000) = 6,400 + 9,600 = 16,000\] IOPS for performance SAS in RAID 6

\[(0.8 \times 2,000) + (6 \times 0.2 \times 2,000) = 1,600 + 2,400 = 4,000\] IOPS for capacity SAS in RAID 6

Common Sense Modifications
As shown above, very few IOPS are being absorbed by the lowest tier. This is as it should be because capacity disks are quite slow. However, in the case of Oracle FS1-2, the granularity of 4 TB capacity disks is large: 80 TB usable. If an enterprise doesn't require a large amount of capacity, then the solution is to add another tray of performance HDDs to deliver sufficient capacity to meet the enterprise's requirements.

Cooling Periods
A simple rule of thumb is that the shorter the cooling period, the higher the skew tipping point. For example, if data cools in 30 days or less, fewer data blocks are hot at any given time than if the data cools over a 60- or 90-day period. As discussed earlier, extended cooling periods are often a result of later processing of the data; either for reports, billing, or analytics. These post-processing routines are typically read-only functions, for which the utilization of the 1.6 TB capacity flash drives of Oracle FS1-2 are ideal. capacity drives read at almost the same speed as performance flash but are about 50 percent cheaper on a capacity basis and hold four times as much data per drive footprint. Cooling periods often are graphically expressed by cooling curves, which are discussed in the following section.

Cooling Curves
Cooling curves often are drawn as smooth curves showing a close to hyperbolic decline in access density over time. Unfortunately, this is not really the case. Transactional data, after the transaction completes, often drops like a stone in access density; but then, at a later point in time becomes active again—during a 30-day billing period, for example. This unexpected behavior confuses the auto-tiering algorithms of most array vendors, leading them to demote data too fast to very low tiers causing subsequent performance problems when the data heats up again. This is not the case with Oracle FS1-2, whose multi-tiered flash architecture incorporates real-life observed behavior into its design.
Both the Telco and financial services access patterns discussed above have extended cooling periods (45 to 60 days or more). A small amount of performance flash can capture the initial transactions, which are write heavy, and Oracle FS1-2 can migrate the completed transactions to capacity flash as it cools. This method delivers flash performance to the applications throughout the cooling period at less cost and footprint than a pure performance implementation. The modified cooling curve in the next section provides a visual example of a data "reheat" scenario.

**Oracle Database Cooling Considerations**

Enterprises that adopted early, first generation auto-tiering arrays with a single tier of flash often experience significant performance issues with Oracle Databases when the array migrated transactional data off of performance flash directly to capacity HDDs. When it is time for a user to run monthly or quarterly reports or business analytics, the performance is terrible. This became such an issue with one particular vendor's solution that the vendor began to tell its customers to disable auto-tiering for transactional databases. In the case of Oracle FS1-2, which utilizes high-capacity flash drives, migrated data still has flash read performance and reporting is not impacted. This is a unique capability of Oracle FS1-2 and should be exploited for transactional databases with extensive post-processing reporting or analytics access.

**Basic 30-Day Cooling Curve**

![Basic 30-Day Cooling Curve](image)

**Figure 2. Illustration of a cooling curve**

**Cooling Curve with Billing Cycle**

![Cooling Curve with Billing Cycle](image)

**Figure 3. Illustration of cooling curve with billing cycle**

Most cooling curves are drawn this way, as a smooth curve; but is that really what happens? Probably not. Data cooling and reheating tend to be much more abrupt and spiky. For example, when you run a quarterly report, it
doesn’t start slowly and then speed up; rather it starts abruptly and needs the data right then and there. Admittedly, there typically are multiple reports, not all run at the same time so there is some ramp time, but it’s not smooth. A better representation of a cooling and reheat cycle might look like this:

![Business Cycle Chart with Re-Heat times](image)

The takeaway here is auto-tiering arrays need to be able to learn fast and must support the ability to bias data migration by the actual business use of the data. The Oracle FS1 does just this via QoS-Plus, the fusion of frequency based auto-tiering with Oracle’s patented QoS. This coupled with multiple tiers of flash can allow for your hot data to be demoted to a high capacity, less expensive read optimized based tier of flash, so when billing reheats happen and/or financial reporting takes place, your applications get the benefit of very fast flash read rates (These reheats are almost all read) without incurring the cost of expensive performance flash.

**Data Growth Rates**

Data growth rates are another useful metric in estimating how much flash is required to capture the majority of hot IO’s. New data by definition is probably hot data. For example, if one has 30 TB of data that is expected to grow at 20 percent per annum, one would add 6 TB a year or ~ (16.4 GB) every day. If the average cooling period is 60 days, then 986 GB (~1 TB) of the data is highly likely to be hot. This equates to 3.3 percent, which is within the working boundaries hypothesis, albeit on the low side.

**The Economics (Business Value) of Storage Media Types**

There is a 30x difference in the cost per GB of storage and a 27x cost per I/O on media available today. The obvious message is it is critical for enterprises to use the most cost-effective media to store data while at the same time meeting business objectives as they relate to performance, latency, and bandwidth.

The table below shows the relative costs of capacity and IOPS on media used in Oracle FS1-2 as of the date of this paper.

**TABLE 2**

<table>
<thead>
<tr>
<th>Media Type</th>
<th>$/IOPS</th>
<th>$/GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>300 GB HDD 10 K RPM SAS 2.5&quot;</td>
<td>$1.92</td>
<td>$1.60</td>
</tr>
</tbody>
</table>
Expressed graphically, the price/performance table would look like this:

<table>
<thead>
<tr>
<th>Product Description</th>
<th>$/GB</th>
<th>$/IOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>900 GB HDD 10 K RPM SAS 2.5&quot;</td>
<td>$3.09</td>
<td>$.86</td>
</tr>
<tr>
<td>4 TB HDD 7,200 RPM SAS 3.5&quot;</td>
<td>$11.20</td>
<td>$.25</td>
</tr>
<tr>
<td>400 GB performance SSD SAS 2.5&quot;</td>
<td>$.41</td>
<td>$7.50</td>
</tr>
<tr>
<td>1.6 TB capacity SSD SAS 2.5&quot;</td>
<td>$1.23</td>
<td>$4.50</td>
</tr>
</tbody>
</table>

$\text{iops} = 27\times \text{Range}$

$\text{g/\text{gb}} = 30\times \text{Range}$

Figure 5: Price Differentials between media types

The only way to leverage these amazing differences in capacity cost and performance cost is with a storage array that has a management framework that can differentiate between hot data, which needs high performance and low $$/\text{IOP}$, and cold data, which needs high capacity and the best $$/\text{GB}$. This combination of ever-changing access density evolves on a continual basis, so the ability to differentiate must be an implicit function of the array. A review of historical access patterns is not enough. Low value, "junk" data may well be promoted to a more expensive tier at the expense of data with a higher business value. This is somewhat similar to trying to drive a car by looking in the rear-view mirror. A possibly interesting experiment on highway 101 in Northern California, or Route 95 on the east coast. Please let us know how it all worked out. Historical access frequency alone is not an answer to lasting business value.

Enterprises must be able to bias inter-tier data movement by data value as well as frequency of reference. Additionally, the type of I/O must be considered: random or sequential, read biased or write intensive. For instance, capacity media is perfectly adequate for highly sequential I/O's, there is no need for flash for this type of access.
pattern. Read intensive data will be happily stored on capacity read-optimized flash which is much lower in cost than performance flash, and is over 4X denser. It should be no surprise by now to realize that the Oracle FS-1 takes into account all of these variables! This is plain, simple economics!

» Single-tier arrays, no matter how fast, cost far too much because they are storing cold data on the most expensive (only) tier.

» Deduplication is a marketing canard as it cannot approach the 30x range of $/GB that the FS1 can deliver today.

» The over-provisioning of low-performance HDDs costs dramatically more in footprint, power, and cooling over the average lifetime of an array (3 to 5 years), not to mention probable higher initial acquisition costs.

The only remaining question is, "Why would enterprises do either of these: Store cold data on flash or over-provision HDDs to beat the problem to death with hundreds of drives? They shouldn’t—period, full stop. Investigate the new sheriff in town, the Oracle FS1-2 flash storage array—the most intelligent, cost-effective disk array available today. It's high on performance where needed and light on budgets when needed (always needed). The embedded data management software, which delivers business value biased auto-tiering (QoS Plus), is included at no extra cost. As a result, there are no "capacity taxes" or “success taxes” and no hidden charges—just good value.

Oracle FS1-2 Overview

The Oracle FS1-2 was designed from the beginning to deliver superior flash performance while also supporting intelligent auto-tiering across both performance and capacity flash SSDs and performance and capacity HDDs (2.5" and 3.5" SAS-2). It can scale up to 912 TB of flash and is designed with enterprise-grade capabilities to provide maximum uptime and data availability:

» <1 second failover capability

» Warm start technology for fast recovery from errors and non-disruptive firmware upgrades

» No single point of failure (SPOF)

» Pre-emptive copy (detects impending disk failures using drive error recovery information and other impending fault indicators developed by Oracle)

» SSD "gas gauges" (SSD usage life remaining)

» T10 Protection Information (T10 PI) for silent data corruption detection

» Thin provisioning and copy services—all included with the software at no additional charge

» Replication options for both synchronous and asynchronous operational modes: fan in/fan out, multi-hop and bi-directional topologies. All offered with application consistent recovery options.4

The Quality of Service Plus (QoS Plus) feature takes into account usage patterns (I/O access frequency), but it also relies on algorithms that identify data as important (high business value) or unimportant (low business value) or in between to manage queues and bias auto-tiering decisions as well as I/O types: random vs. sequential and read vs. write operations. As a result, it drives adaptive, automated data movement to the most cost-effective media from a $/IOP and $/GB standpoint using a sub-LUN tiering approach. Low-value data is placed on lower-cost HDD media while important, high value, hot data is placed on performance media.

Additionally, the Oracle FS1-2 system has the support of Oracle Database features such as Hybrid Columnar Compression, which delivers up to 30x efficiency in compression ratios that reduce the need for additional physical storage. It should be noted that if you combine HCC’s 30X compression with the 30X differential in $GB costs between performance media and capacity media, the overall user cost savings come to an incredible 900X reduction in incurred storage costs. Just ask yourself this simple question: Knowing that most all data has a high

4 Replication software is the only licensed software on the FS1. All other data management functions are included at no additional charge.
skew value, would you prefer to have 100% of your data on high cost flash media even with 5X deduplication, promised by many All Flash Array (AFA) vendors but rarely achieved, or 95% of your data stored on media where you could realize 900X lower costs per GB with HCC (30x price advantage with 4 TB HDD*30X HCC compression) and 5% of your data on flash at 1/3rd “list price” utilizing Oracle’s Advanced Compression Option (ACO) and, for a bonus round employ Automatic Data Optimization (ADO) to allow the data base to automatically invoke the best HCC compression scheme as data ages. The math is obvious; the latter case delivers fully automated data management and, as such, much better business value. Only from Oracle.

Plug-ins are available for Oracle Enterprise Manager and the Oracle VM Storage Connect feature, in addition to support for Automatic Data Optimization, a feature of Oracle Database, also contribute to additional IT and human efficiency. The Oracle FS1-2 system datasheet provides additional information.

The fusion of multiple Oracle delivered features delivered in combination with each other are the basis of the Oracle Red Stack Advantage.
Conclusion

The empirical data discussed in this paper, along with the IOPS calculation methodology, should enable storage decision-makers to configure an Oracle FS1-2 auto-tiering array with a high degree of confidence. It is designed to protect enterprise investments over time, to meet or exceed user expectations and not to cause performance problems by over-aggressive data demotion. Capacity on any tier can be added on the fly, and it can be utilized automatically by the Oracle FS1-2 flash storage system with no impact to the application or requirement for manual inter-tier data migrations.

Based on multiple real-life observations and mathematically sound estimation methods, it is reasonable to estimate the range of data likely to be hot at any given point in time is a low of ~3 percent to a high of ~6 percent. This range represents a good starting point from which to refine final array configurations.

Placing all of an enterprise’s data on the same tier—flash, high-performance HDDs or any other option—is almost certainly a poor economic decision. Enterprises that drastically over-provision hard drives or choose the all-flash route end up paying far too much for performance and miss-allocate expensive resources in an attempt to solve problems for which the technology is not well suited.

Additionally, QoS Plus applies a business value and I/O structure bias to frequency-based tiering. This is an industry-first capability, which aligns data location and resulting performance characteristics with business needs versus simple frequency of use. Multiple flash tiers allow for cost-effective flash performance to be delivered consistently to applications with extended or non-uniform cooling periods.

The Oracle FS1-2 flash storage system is the most intelligent, cost-effective auto-tiering array in the market. When using the FS1-2, enterprises know what they are paying for upfront and pay no “growth or success taxes” ever.

When you have the capability to develop hardware with the team who write the software that will execute on it, it becomes fairly obvious that the resulting whole will be far greater than the sum of a bunch of disparate parts. The RED STACK, only from Oracle.