We live in a world where the status quo has been ever faster processor, memory and network speeds, yet no corresponding improvement in storage performance. In fact, from a performance-per-GB perspective, storage has been getting slower at the same rate that processors have been getting faster! The result is that we have evolved an entire ecosystem of coping mechanisms to try and enable storage to keep up with the demands of our applications.

Flash is not disk. Flash memory arrays from Violin Memory require you to take a new look at everything you thought you knew. What if you didn’t have to cope with slow storage anymore? What would you be able to accomplish?

Here we look at a number of compromises which no longer need to be made:

### 1. Capacity Over-Provisioning

How much of your storage system can you actually use?

- Legacy storage systems boast large capacities which cannot be used without causing performance problems
- This is caused by the imbalance between the capacity of disk and the number of I/O operations it can service per second (IOPS)
- A typical 600GB SAS hard drive can perform less than 200 IOPS but modern applications require many orders of magnitude more
- To meet the application requirements, large numbers of disks must be used, increasing operational costs and yet wasting capacity
- How much more capacity must be wasted to cope with future demand?
- For high-performance applications the latency of disk is unacceptable, so vendors have to resort to “short-stroking”, when only the outer 10% of the disk platter is used
- The cost of this compromise is that between 75% and 90% of the disk capacity is wasted just to remove one or two milliseconds of latency
- This is stranded capacity – it is available but cannot be used
- The required number of disks just increased by 10x and yet none of that capacity is available for use!

Violin Memory flash Memory Arrays provide storage at the speed of memory, with >1 million IOPS available at sustained low latency from a single array, unleashing applications and allowing increased application performance while reducing operational costs.

### EXAMPLE:

A 3TB database performs 100,000 IOPS during the online day. The disk vendor needs to use 500 disks to meet this requirement, giving a total of 300TB – yet the remaining capacity cannot be used without performance degrading, so 99% of capacity is wasted. Even in the best case where 90% of the I/O can be satisfied in memory, 50 disks would be required totalling 15TB of capacity – a 500% overprovision.
Caching
Is your storage performance based on luck?

• All disk storage vendors attempt to mask the relatively slow performance of their storage systems with the use of a cache
• The cache allows for some “lucky” I/O operations to complete with faster response times
• However, “unlucky” operations experience the extra penalty of a “cache miss”, looking unsuccessfully for the data in cache before reading it from the underlying storage system
• As a result, performance is unpredictable, especially when operations such as backups and batch jobs clear the cache of useful data

The idea of an array cache simply does not make sense when you can have the entire data set contained in ultra-fast low-latency flash memory. Performance is predictable and guaranteed, no matter what the workload.

Disk Protection and Data Layout
How many combinations of disk options do you need to consider when designing a storage solution?

• RAID level 0, 1, 4, 5, 6 or 10 protection schemes? SAS, SATA, Fibre-Channel or SSD media type? What would be the optimal stripe size for this system?
• Does one design fit your requirements or do you need to design a solution for each data-type? Are logs, data, indexes etc all best served by different strategies? If the storage is shared, are the designs compatible?
• How much time needs to be spent looking for the best solution? And at what cost?
• Different RAID levels were designed to allow you to trade-off capacity with performance to achieve the most economical solution. When performance is virtually unlimited, there is no choice to make.

With Violin Memory’s patented vRAID technology (designed specifically to enhance NAND flash system performance) all applications receive maximum performance and spike-free latency with full hardware-based RAID data protection.

I/O Scheduling
How much CPU are you sacrificing to cope with the performance of storage?

• Most operating systems use an I/O scheduler to manage the order in which I/O operations are submitted to storage
• These complex algorithms re-order I/O operations so that the movement of disk heads (known as a “seek”) is reduced
• The compromise is therefore to spend CPU cycles in order to reduce seek time and the associated impact on performance

With Violin Memory flash Memory Arrays the idea of a “seek time” no longer exists and I/O operations can be executed with ultra-fast response times. This means CPU resource no longer needs to be wasted and can be freed for use by applications and users to get the best performance.
Automatic Tiering
Is your storage always one step behind the application?

- Some storage vendors attempt to further mitigate performance problems by providing multiple “tiers” of storage offering different performance characteristics.
- Automatic software attempts to migrate data depending on its level of use, with “hot” data on the faster tiers.
- The simple but unavoidable problem with this approach is that past behaviour is not indicative of the future.
- Any irregular usage pattern can have a vast and detrimental on-going effect on performance.

Disk Arrays cannot accurately move data because they do not understand its relevance to the application. Flash Memory Arrays from Violin Memory do not need to migrate data because all I/O operations are serviced with sustained ultra-low latency.

Example:
Every database makes use of a data cache, where the “hottest” blocks remain in memory. As a result, the persistent copies of these blocks on disk can appear to be “cold”, causing the automatic tiering software to move them to slower storage. A user then changes the information in one of these hot blocks, causing the database to write the new data to disk. The blocks are now located on slow storage causing poor update times and resulting in locking issues as the database waits for the write to complete.

Manually Distributing I/O
Are you adding complexity to achieve acceptable performance?

- A common tactic to alleviate performance issues is the manual distribution of data based on knowledge of its use.
- Examples of this include the manual archiving of historical data, or the use of SSD or PCIe flash cards for temporary tablespaces.
- Manual distribution carries the challenge of complexity as well as the overhead of constant monitoring to ensure the configuration remains optimal.

When all file types can share ultra-fast low-latency flash storage from Violin Memory, why choose complexity over simplicity?

Over-Indexing
Does your database contain more indexes than data being indexed?

- Databases with high I/O requirements often exhibit performance problems due to the high latency associated with reading random blocks from legacy storage.
- Indexes are often the biggest problem because by nature index reads are random.
- A common tactic of the DBA or application architect is to over-index tables in order to try and reduce the number of blocks which must be read from storage.
- The overhead required to maintain numerous indexes has significant impact on any insert, update or delete operations.

Violin Memory flash Memory Arrays represent a new level of performance for random I/O, with latencies that are orders of magnitude faster than disk systems. With flash memory there is no need to resort to “index trickery” to alleviate the cost of reading data from storage.
**Foregoing Real Time Data**
Why make do with stale information when you can work in real time?

- Many database systems use summary tables to calculate point-in-time snapshots of data
- This summary information cannot be calculated more frequently due to the impact on performance
- Disk arrays experience exponentially-increasing response times under heavy load

The extreme capabilities of Violin Memory flash Memory Arrays allow for huge volumes of data to be read in short spaces of time with minimal impact to other users. Maybe those summary tables can be calculated every five minutes instead of every six hours? Or maybe the data can be calculated in real-time...

**EXAMPLE:**
A billing system uses “materialized views” to show the total and average daily invoice values for each region to the finance team. Due to the volume of data in the invoice table, the legacy storage system is unable to continuously calculate these values in real-time, so the "snapshot" of the values is calculated every six hours. As a result, the information available to the finance team is constantly out of date.

**Offloading Reports and Analytics**
Is your business intelligence system always stuck in the past?

- Many organisations use reporting or decision-support databases in order to offload the overhead of reporting and analytic tasks
- These databases have data "bulk loaded" from the main system so that aggregation and calculation routines can be run without affecting live operation
- The additional operational and licensing costs for these databases can be substantial, yet the compromise is often that the data is stale – by the time it is loaded it may already be out of date
- Synchronising the data in real time with the use of products such as Oracle GoldenGate or Active Data Guard incurs a further substantial licensing cost and adds complexity

By removing the I/O bottleneck the option to run reporting on the live database now becomes realistic. This offers the opportunity for a far more simple design, yet one which allows for large savings in licensing costs and gives all users access to real time data. Databases running on Violin Memory now have all of the benefits of an in-memory database.

**Coping**
Why compromise is no longer the only option...

- Many users of legacy storage devices are unaware of just how much effort and resource is required to mitigate the performance of these systems
- Many of the strategies listed here are used without consideration for the overhead they incur, simply because “it has always been that way”
- The time has come to stop compromising over the performance of legacy technology and start considering what the wasted effort and resource could be spent doing instead

Flash Memory Arrays from Violin Memory offer the prospect for storage to cease being a problem and start being an opportunity. For many years customers and software vendors have attempted to avoid accessing storage where possible to retain the speed of performing operations in memory. With flash Memory Arrays from Violin Memory this is no longer necessary, because storage is now performing at the speed of memory.