WHITE PAPER

Data Center
Memory and Storage

RAID Performance Analysis on Intel® VROC

Exploring the mathematics behind RAID performance benefits and how they relate to Intel® VROC.

Table of Contents

Purpose

This white paper provides an analysis of RAID performance and describes the mathematics behind RAID performance, as well as the specific RAID implementations of Intel® Virtual RAID on CPU (Intel® VROC) and how operating systems, storage devices, and other components impact performance output.

Scope

RAID storage solutions aim to provide users with an appropriate combination of data protection and performance acceleration. Each RAID level processes storage I/O in a different manner and stores data in a specific pattern across a set of RAID member disks. Our goal is to highlight those storage patterns for RAID levels 0/1/10/5 and explain how each pattern affects the performance of the storage solution. By understanding the underlying I/O process, one can understand the theoretical performance maximum for each RAID level. We will explore how Intel VROC implements these RAID levels in an actual product and how Intel® VROC's performance compares to the theoretical maximums laid out. In the future, as this document continues to evolve, we will explain why any performance deltas exist between Intel VROC and the theoretical maximums caused by the Intel VROC RAID engine, OS limitations, SSD architecture, and more.

Target Audience

Data Center Administrators, Architects, and Managers; Storage Performance Testers; Server Platform Manufacturers

Terms and Symbols

Throughout the rest of this section, the following terms and symbols will be used:

- N: Total number of drives in a RAID volume.
- Disk IOPS: Number of maximum I/O operations per second for a single RAID member drive. **Note:** All formulas apply to IOPS and bandwidth (one can easily be converted to the other).
- Disk X/Y mixed IOPS: Number of maximum I/O operations per second for a single drive, which is a total of read and write IOPS, for a mixed workload of X% read and Y% of write. **Note:** All formulas apply to IOPS and bandwidth (one can easily be converted to the other).
- RAID IOPS: Number of maximum I/O operations per second for a RAID volume. **Note:** All formulas apply to IOPS and bandwidth (one can easily be converted to the other).
- SS: RAID volume Strip Size

- Sequential I/O: I/O performed by a single thread, where each subsequent I/O request LBA is adjacent to the previous one (all I/Os are sent in a sequence in regards to the LBA).
- Random I/O: I/O performed by a single or multiple threads, where all the I/O requests' LBAs are randomly and uniformly distributed across the entire logical address space of the device (or a RAID volume).
- QD: Queue Depth per worker
- WC: Worker Count
- LBA: Logical Block Address
- XOR: Exclusive OR, a standard Boolean logic operator that compares two bits and returns 1 if one, and only one, of the inputs is 1. It is often used in parity calculations and cryptography.
- VMD domains: Intel® Volume Management Device, new hardware in the root complex of Intel® Xeon® Scalable Processors for connecting and managing NVMe* SSDs
- VROC: Intel® Virtual RAID on CPU, the NVMe* SSD RAID solution provide by Intel, also the RAID technology being evaluated in this document

RAID Mathematic Performance Model

RAID mathematical models define the maximum theoretical performance of RAID systems, not including limitations imposed by hardware or software. In fact, some of these numbers may not be achievable in real world settings due to hardware or software limitations. In such cases, an explanation will be provided later in this document.

Summary Table

RAID level	Read	Write
RAID 0	$N \times Disk\ IOPS$	$N \times Disk\ IOPS$
RAID 1	$2 \times Disk\ IOPS$	$Disk\ IOPS$
RAID 10	$4 \times Disk\ IOPS$	$2 \times Disk\ IOPS$
RAID 5 (3 drives)	$N \times Disk\ read\ IOPS$	$\frac{3 \times Disk\ random\ write\ IOPS}{2 + \frac{Disk\ random\ write\ IOPS}{Disk\ random\ read\ IOPS}}$
RAID 5 (4 or more drives)	$N \times Disk\ read\ IOPS$	$\frac{N \times Disk\ random\ write\ IOPS}{2 + 2 \times \frac{Disk\ random\ write\ IOPS}{Disk\ random\ read\ IOPS}}$

Strip and Stripe

Figure 1 below shows the standard naming conventions used in regards to the data layout on a RAID volume.

Model Assumptions

For simplification, the following assumptions has been made for the model:

1. The RAID engine does not reorder the I/O requests (sequential workload is not randomized in any way)
2. For random workloads:
    a. The I/O size is less or equal to SS
    b. SS is equal to or multiple of I/O size
    c. I/O logical address (LBA) is I/O size aligned
3. For the 'Disk IOPS' measurement, the workload QD and WC are adjusted in a way that the drive performs at its maximum IOPS
4. Likewise, for the 'RAID IOPS' measurement, the workload QD and WC should be set so that all the RAID member drives perform at their maximum IOPS. The QD and WC values used for this measurement may be different than those used in the 'Disk IOPS' measurement, but the I/O size and other parameters should be the same, unless stated otherwise.

RAID 0

RAID 0 is a simple distribution of data across all the member drives.

The universal formula for RAID IOPS, both random and sequential, both read and write, is:

RAID IOPS = N X Disk IOPS

All RAID member drives execute the I/O concurrently and the workload is evenly distributed to member drives. Consequently, there is no RAID overhead work to negatively impact performance.

Note: In Figure 2 below D1,1 D1,2 and D1,3 can all be read/written at the same time, thus performing three I/Os in the time it usually takes to perform one.

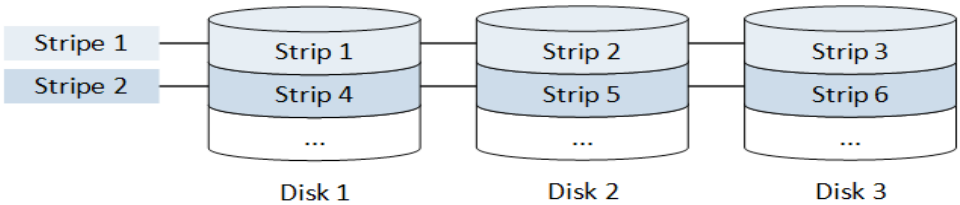

Figure 1: Naming Conventions related to RAID data layout

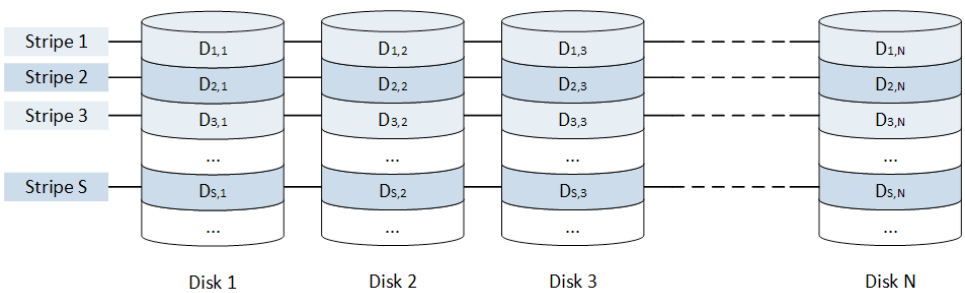

Figure 2: RAID 0 data layout

RAID 1

RAID 1 is a simple copy of data on two disks. The read I/O is distributed across the two member disks.

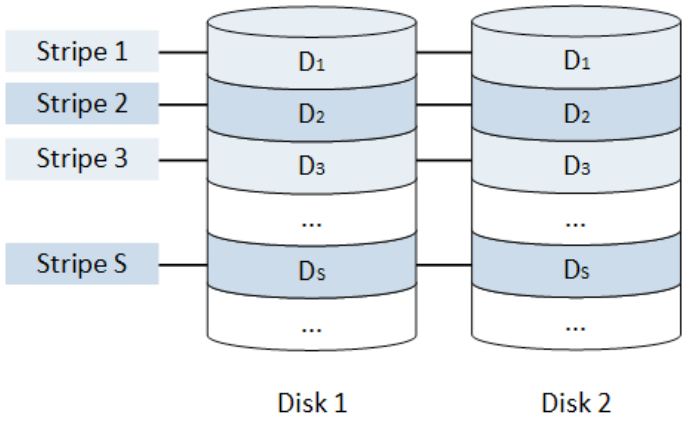

Figure 3: RAID 1 data layout

The formula for RAID IOPS, both random and sequential is:

RAID read IOPS = 2 X Disk read IOPS

Note: In Figure 3 above D1 and D2 are read at the same time.

The formula for RAID IOPS, both random and sequential is:

RAID write IOPS = Disk write IOPS

A copy of data is written to each RAID member drive; no performance improvement is achieved.

RAID 10

RAID 10 is a nested RAID level with a RAID 0 on top of two or more RAID 1 volumes.

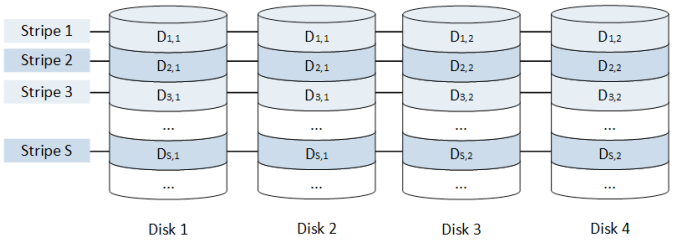

Figure 4: RAID 10 data layout

Read I/O is distributed across all member drives.

The formula for RAID IOPS, both random and sequential is:

RAID read IOPS = N X Disk read IOPS

The reason for this is that read I/O can be distributed across all four member disks.

Example: D1,1 and D2,1 can be read at the same time as D1,2 and D2,2. These four I/Os can be completed in the time it normally takes to do one.

The formula for RAID IOPS, both random and sequential, for write, is:

RAID write IOPS = N/2 X Disk write IOPS

Note: In Figure 4 above D1,1 and D2,1 are read at the same time as D1,2 and D2,2. These four I/Os can be completed in the time it normally takes to complete one.

The formula for RAID write IOPS, both random and sequential is:

RAID write IOPS = N/2 X Disk write IOPS

Note: In Figure 4 above D1,1 and D1,2 are written at the same time, each with a duplicate copy.

RAID 5

RAID 5 (see Figure 5 below) is a distribution of data across all the member drives with one strip of parity for each of the stripes. The parity is an XOR of all other data strips within a stripe and is equally distributed across all the drives.

For read I/O, RAID 5 is similar to RAID 0 and the formula is:

RAID read IOPS = N X Disk read IOPS

All the drives perform I/O concurrently and the parity does not affect the performance in any way – parity is overlooked. The only difference between RAID 0 and RAID 5 is that RAID 5 sequential read I/O, from the perspective of a RAID volume, is no longer sequential from the perspective of a RAID member drive, because parity is overlooked. If a drive uses some kind of sequential I/O detector and pre-fetching mechanism to read-ahead the data, this mechanism might pre-fetch both the data and parity, which may introduce unnecessary overhead.

Note: In Figure 5, stripe 1, D1,1, D1,2 and D1,3 can be read at the same time. P1 is overlooked, but D2,N-1 can be read, meaning four strips can be accessed at the same time.

For write I/O, there are several cases described in the following sub-sections. There is also a section describing read-others and read-modify-write operations, which is helpful in understanding the formulas.

Read-Others and Read-Modify-Write

There are two methods of performing write operations to a RAID 5 volume. For each I/O operation, a RAID engine selects the optimal method (the method which results in fewer I/O operations).

Below is the formula to calculate a new parity value for the stripe k in the case where Dk,1 is being updated:

$$P_{k\,new} = D_{k,1\,new} \otimes D_{k,2\,old} \otimes D_{k,3\,old} \otimes \ldots \otimes D_{k,n\,old}$$

In order to calculate new parity, all the data strips are required. If any of them are being updated, the new data is already in RAM. All other strips need to be read from the drives, generating additional read requests which are overhead. This algorithm of parity calculation is called the **read-others** method.

Let's consider another formula for the same case as above (Dk,1 update).

$$P_{k\,new} = D_{k,1\,old} \otimes D_{k,1\,new} \otimes P_{k\,old}$$

The resulting parity is the same as calculated using the previous formula, but the data used in the equation is different. Old data and old parity is read and XOR-ed together with the new data. There is no need to touch the other data strips. This method is called **read-modify-write**.

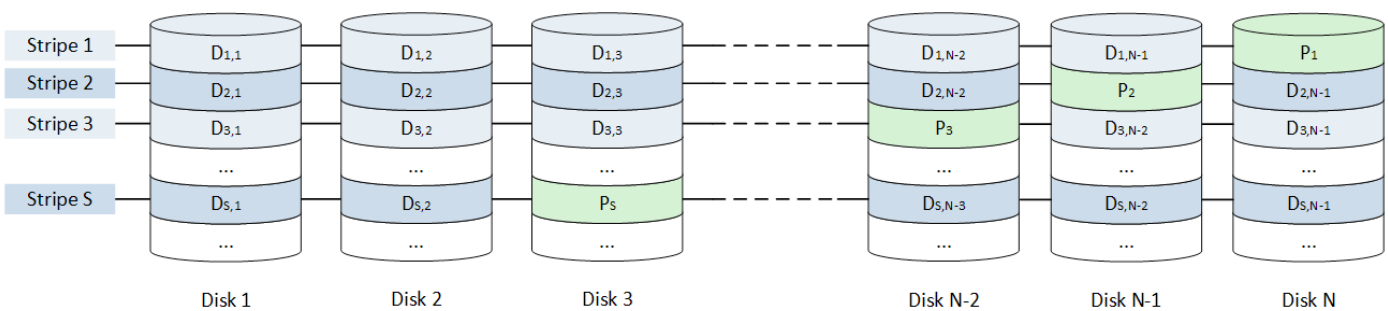

Figure 5: RAID 5 layout

Small, Random I/O

In order to perform a write operation to a RAID 5 volume, both data and parity need to be updated. This requires either read-others or read-modify-write sequence (see Read-others and read-modify-write above), depending on which one results in fewer I/O operations. Typically, read-others will only be applied to a 3-drive volume.

This paper describes two methods on how to calculate RAID 5 write IOPS.

Method 1 - based on a single disk maximum read and write IOPS

This method is based on the assumption that a drive has a 'budget' of I/O operations, which is expressed as either 'Disk write IOPS' or 'Disk read IOPS'. Based on this, a read operation can be translated to an associated write IOPs cost.

$$RAID\ random\ write\ IOPS\ (3\ drives) = \frac{3 \times Disk\ random\ write\ IOPS}{2 + \frac{Disk\ random\ write\ IOPS}{Disk\ random\ read\ IOPS}}$$

$$RAID\ random\ write\ IOPS\ (4\ or\ more\ drives) = \frac{N \times Disk\ random\ write\ IOPS}{2 + 2 \times \frac{Disk\ random\ write\ IOPS}{Disk\ random\ read\ IOPS}}$$

The formulas are based on a principle that all the RAID member drives perform write I/O in parallel (numerator), but there is a performance overhead for each I/O, which consists of additional read and write operations (denominator) that are not directly committing the initial I/O data to storage. The total overhead impact is a function of the number of additional drive write operations plus the number of read operations.

Method 2 – based on a single disk read-write mixed IOPS

This method is more precise, consequently it requires more information about the drive than just the maximum read or write IOPS. It is based on a measurement of the drive in the exact conditions, under which the drive performs in RAID.

$$RAID\ random\ write\ IOPS\ (3\ drives) = \frac{3 \times Disk\ 33/67\ mixed\ IOPS}{3} = Disk\ 33/67\ mixed\ IOPS$$

$$RAID\ random\ write\ IOPS\ (4\ or\ more\ drives) = \frac{N \times Disk\ 50/50\ mixed\ IOPS}{4}$$

The main principle in this method is that there are N drives executing the I/O in parallel (numerator) and the overhead (denominator) is expressed in total number of I/O operations (reads or writes) required for a single RAID write operation.

Sequential I/O, Partial Stripe Write

Typically, sequential workloads involve relatively large I/O requests (e.g. 128KB in size). The I/O request is usually larger than the RAID strip size, which makes the theoretical performance hard to calculate accurately (for cases other than the full stripe write).

Additionally, because of the RAID 5 data layout—the necessity to update parity strips along with the data—and due to the multithreaded architecture of some RAID engines, the sequential nature of the I/O becomes random from the perspective of the RAID member drives. This means that a measurement of a single drive, with a sequential workload, can no longer be a reference measurement used to calculate the performance of RAID. Performance of RAID 5 sequential writes can also vary depending on the RAID engine implementation.

For simplicity, the following assumptions are made:

1. The RAID engine can distribute the I/O requests in a way that, despite the per-stripe parity locks (causing blocking of some of the requests), each RAID member drive can be fully saturated. In other words, parity locks will be applied if two or more write I/O requests land in the same stripe, and therefore attempt to update the same parity strip; in such a case only one of these requests can be executed at a time.
2. The workload queue depth is large enough that, despite the per-stripe parity locks, each RAID member drive is supplied with the number of concurrent I/O requests, and thus can saturate that drive.
3. The formulas describe the minimum performance. In practice (especially when the strip size is small), full stripe writes or read-others algorithm may occur, both of which can improve the performance.

$$RAID\ sequential\ write\ IOPS\ (3\ drives) = \frac{3 \times Disk\ random\ write\ IOPS}{2 + \frac{Disk\ random\ write\ IOPS}{Disk\ random\ read\ IOPS}}$$

$$RAID\ sequential\ write\ IOPS\ (4\ or\ more\ drives) = \frac{N \times Disk\ random\ write\ IOPS}{2 + 2 \times \frac{Disk\ random\ write\ IOPS}{Disk\ random\ read\ IOPS}}$$

Note: Sequential RAID performance is calculated based on drive random performance, because the workload from the drive perspective is no longer sequential (due to multiple parity updates). Please also note that the formulas are identical to the formulas used to calculate the performance of small, random I/O. The alternative (more precise) formulas can be used here as well.

Sequential I/O, Full Stripe Write

RAID 5 full stripe write takes place if the following formula is true:

$$RAID\ I/O\ size = (N\text{-}1)\ X\ SS$$

In the full stripe write case, there is no need for any additional read operations since the parity is pre-calculated in RAM. There is however overhead related to writing the parity. The total capacity of the parity in a RAID 5 volume is equal to a single member drive, which implies a write bandwidth drop by a factor of a single drive bandwidth.

$$RAID\ sequential\ write\ IOPS\ (full\ stripe\ write) = (N\ X\ Disk\ IOPS) - Disk\ IOPS = (N\text{-}1)\ X\ Disk\ IOPS$$

This formula assumes that any overhead related to I/O fragmentation—based on the strip size, due to the RAID 5 data layout—is negligible.

OS Performance Impacts on Intel VROC

In some cases, Intel VROC performance does not meet the theoretical limits explained in the section above. Sometimes this is due to the impact that the OS has on Intel VROC.

Windows* Limitations

Intel VROC is a kernel storage driver based on a storport-miniport architecture:

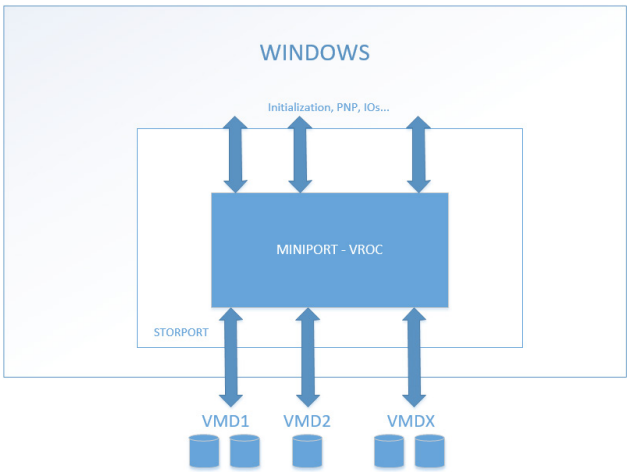

Figure 6: VROC in Windows* storage rack

A single Intel VROC Driver instance is loaded on top of all VMD domains. This Intel VROC Driver handles communication with SSDs connected to each domain, and has the capability to merge those SSDs into RAID volumes. RAID logic is implemented on the software side and is realized by the Intel VROC RAID Engine. In Windows, current Intel VROC RAID Engine architecture has limited efficiency and in some cases can limit theoretical maximum RAID performance.

For maximum theoretical performance when millions of operations per second occur, the performance of the Intel VROC driver is negatively impacted by the number of shared structures that cannot be accessed in parallel. This results in increased lock contentions and finite performance scalability. Due to Intel VROC RAID Engine efficiency IOPS numbers are currently limited to:

- ~1M IOPS for single RAID volume
- ~1.4M IOPS for multiple RAID volumes

Intel VROC RAID ENGINE Adjustments

The Intel VROC RAID Engine is multithreaded; by default there are 10 active threads. The optimal number of Intel RAID Engine threads may vary for different setups and different workloads. This value is adjustable and can be changed by modifying the following registry key:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\iaVROC\Parameters\Device\Threads

Reducing the number of active threads will limit maximum performance. Increasing the number of active threads may improve overall maximum performance, especially for workloads generated to multiple RAID volumes in parallel. Users should not set the threads number to a value higher than the number of physical cores on the first socket of a target platform. To apply the changes a platform reboot is required.

Conclusion

Understanding RAID fundamentals is critical to properly evaluating a storage solution. Each RAID level offers a tradeoff between redundancy, performance, and cost, and this document can be referenced to pick the right RAID level for specific storage needs. Furthermore, Intel VROC is a hybrid RAID solutions that delivers RAID functionality to the NVMe ecosystem. By leveraging NVMe SSDs connected directly to the CPU, Intel VROC enables optimized RAID configurations that approach many of the theoretical maximums explored above. Ask your platform provider about Intel VROC to unleash the value of NVMe SSDs today!

To learn more, visit www.intel.com/vroc