

Comprehensive Guide to Configuring Proxmox VE with SupremeRAID™ for Enhanced Virtualization Performance

May 2024

Table of Contents

Purpose of the Guide.....	4
Introduction.....	4
Why People Choose Proxmox	4
Overview of Proxmox and SupremeRAID™	4
Understanding Hypervisors.....	5
Types of Hypervisors	5
Performance.....	6
Security	6
Use Cases	6
SupremeRAID™ on Hypervisor Type 1.....	7
Recommended Settings.....	8
Bus/Controller Configuration.....	8
SCSI Controller	8
IO Threads.....	9
Cache Settings	9
AIO Type.....	9
Disk Format	9
Integrating SupremeRAID™ with Proxmox VE.....	10
Install Proxmox VE.....	10
Install SupremeRAID™ Drivers.....	10
Driver Installation Instructions.....	11
Download Pre-Installer	12
Download the Latest Driver Package.....	12
Post-Installation Setup.....	13
SupremeRAID™ Configuration	13
Filesystem Setup	15
Creation of Virtual Machines (VMs).....	19
Performance Comparison	20
Baseline Benchmarks Among Various RAID Configurations	20

Aggregate FIO Performance Summary (Random IO) (Bare Metal vs. SupremeRAID™ vs. MDADM vs. ZFS).....	20
Aggregate FIO Performance Summary (Sequential IO) (Bare Metal vs. SupremeRAID™ vs. MDADM vs. ZFS).....	21
Virtual Machine Performance	21
Aggregate FIO Random Read Performance Across Varied VM	22
Aggregate FIO Random Write Performance Across Varied VM.....	22
Aggregate FIO Random Read (Rebuilding) Performance Across Varied VM	23
Aggregate FIO Random Write (Rebuilding) Performance Across Varied VM.....	23
Aggregate FIO Sequential Read Performance Across Varied VM	24
Aggregate FIO Sequential Write Performance Across Varied VM	24
Aggregate FIO Sequential Read (Rebuilding) Performance Across Varied VM.....	25
Aggregate FIO Performance Summary (Random IO) 8VM (6 Disk) vs 48VM (1 Disk)	25
Aggregate FIO Performance Summary (Sequential IO) 8VM (6 Disk) vs 48VM (1 Disk)	26
Conclusion.....	26
Appendix.....	27
Test Environment System Information.....	27
SupremeRAID™ RAID Configuration.....	28
MDADM RAID Configuration and Settings	28
ZFS Configuration.....	29
Virtual Machine Configuration	30
FIO Script for VM Testing	30
Additional Resources.....	34

Purpose of the Guide

This guide is tailored to aid IT professionals, system administrators, or technology enthusiasts in smoothly integrating the SupremeRAID™ driver with Proxmox VE. The objective is to give users easy-to-follow, detailed steps for implementing and using SupremeRAID™ by Graid Technology.

Proxmox offers different types of data stores. However, SupremeRAID™ allows Proxmox to maximize local storage, ultimately boosting the performance of your virtual machines. Upon completing the procedures detailed in this guide, customers will have established a strong and effective virtualization setup, harnessing the strengths of both Proxmox and SupremeRAID™ technologies.

Introduction

Why People Choose Proxmox

Proxmox VE stands out with its flexible network configuration options for simple setups using a bridged network model. This model ensures seamless integration of virtual environments with the external network. It's an open-source platform, eliminating the need for a license. Customers often see Proxmox as a viable alternative to VMware Virtualization. It offers APIs, automation, HA solutions, and live migration, making it a comprehensive and cost-effective choice.

Overview of Proxmox and SupremeRAID™

Server performance and reliability are crucial for enterprise operations in today's data-driven business environment. Traditional RAID or other software tools often fall short when it comes to data storage and protection in the face of modern data demands. That's where SupremeRAID™ comes in—a revolutionary technology that overcomes the limitations of traditional RAID.

SupremeRAID™ maintains ultra-high service performance even during RAID rebuilding, with outstanding performance metrics such as 28M IOPs in random reads, 1M in random writes, 260 GB/s in sequential reads, and 100 GB/s in sequential writes in optimal status. It's the ideal solution for businesses that require high performance and availability along with the ability to handle large data volumes and offer unprecedented performance for demanding applications.

But that's not all—SupremeRAID™ eliminates the need for traditional Battery Backup Units (BBUs), reducing maintenance costs and complexity while enhancing overall system reliability. This innovative technology is set to transform data storage and protection in the modern enterprise context.

Graid Technology offers a comprehensive performance analysis with SupremeRAID™ as compared to traditional RAID technologies. Additionally, Graid Technology will include usable performance reports for the Proxmox environment. SupremeRAID™ can be used in many Hypervisors type 1 for virtualization environments (example list below).

The adoption of SupremeRAID™ comes with several advantages, such as:

- faster performance
- greater reliability
- improved efficiency

These advantages and many others make SupremeRAID™ a must-have for modern enterprise customers.

Understanding Hypervisors

Hypervisor 1 and Hypervisor 2 both refer to software that enables virtualization, allowing multiple operating systems to run on a single physical machine. However, there are different types of hypervisors based on their architecture and how they manage virtual machines. Here are some key differences between Hypervisor 1 and Hypervisor 2:

Types of Hypervisors

- **Hypervisor 1** is a Type 1 or bare-metal hypervisor that runs directly on the server's physical hardware. It does not require an underlying operating system and provides direct access to hardware resources for virtual machines (examples: KVM, RHV, Proxmox, XenServer, Citrix Hypervisor, and XP-NG).
- **Hypervisor 2** is a Type 2 or hosted hypervisor that runs on top of a host operating system. It relies on the host operating system to manage hardware resources and provide services to virtual machines (examples: Parallels, Oracle VirtualBox).

Performance

- **Hypervisor 1** typically offers better performance than Hypervisor 2 because it has direct access to hardware resources, leading to lower overhead and improved efficiency.
- **Hypervisor 2** may introduce additional overhead due to the host operating system layer, which can impact performance compared to Hypervisor 1.

Security

- **Hypervisor 1** is often considered more secure than Hypervisor 2 because it operates independently of the host operating system, reducing the attack surface and potential vulnerabilities.
- **Hypervisor 2** may be more susceptible to security risks since it relies on the host operating system for resource management and services.

Use Cases

- **Hypervisor 1** is commonly used in enterprise environments and data centers where performance and security are critical factors.
- **Hypervisor 2** is often used in testing and development environments or for desktop virtualization on personal computers.

Overall, the choice between Hypervisor 1 and Hypervisor 2 depends on your specific requirements and is determined by how customers want to test and evaluate their solution. Hypervisor 1 is a better option for performance and security as a use case. It is essential to evaluate these factors when selecting the appropriate hypervisor for virtualization needs.

SupremeRAID™ on Hypervisor Type 1

Enhanced Storage Performance

SupremeRAID™ technology offers a remarkable boost in storage speed and efficiency, making it an unparalleled choice for intensive virtualization tasks. This guide provides an in-depth tuning manual and showcases a compelling performance comparison between (Linux RAID) MDADM and SupremeRAID™.

In rigorous and intense testing scenarios with 8 concurrent virtual machines (VMs) to simulate different workloads in VM environments, SupremeRAID™ consistently outperforms MDADM and can deliver over 3 times the efficiency in Random I/O operations. More impressively, in terms of write throughput, SupremeRAID™ showcases a staggering performance advantage, providing more than 10 times the output of MDADM. This superior performance underlines SupremeRAID™'s capability to handle high-demand, data-intensive environments with exceptional speed and reliability.

Scalability

This setup allows for easy scaling of storage resources, accommodating growing data needs without compromising performance.

Reliability and Redundancy

The use of SupremeRAID™ is pivotal in our setup. It provides high reliability and redundancy, thereby ensuring data integrity and significantly reducing the risk of data loss.

This approach enhances the capabilities of a Proxmox VE setup and ensures that users can efficiently manage and scale their virtualization infrastructure.

Recommended Settings

Customers can configure their environment in many ways;
Graid Technology's recommendations are as follows:

Parameter	Description	
Default In 7.4	Recommended Setting	
-----	-----	-----
Bus/Controller	Emulate storage controllers	n/a VirtIO Block
SCSI Controller	SCSI emulation driver type	n/a scsi-virtio-single
IOThreads	Provides dedicated threads for disk I/O	disabled enabled
AIO Type	System interface used to Asynchronous I/O	io_uring native
Disk format	Target format for file storage	qcow2 qcow2 or raw
Cache	drive's cache mode	none none

Detailed instructions for those configurations are as follows:

Bus/Controller Configuration

- **Option:** VirtIO Block
- **Purpose:** Enabling "VirtIO Block" to integrate the "scsi-virtio-single" controller, which enhances performance by ensuring each disk has its dedicated VirtIO SCSI Single controller.

SCSI Controller

- **Option:** scsi-virtio-single
- **Explanation:** This configuration allows each disk to operate with its own SCSI controller, optimizing performance through dedicated resources.

IO Threads

- **Setting:** Enable
- **Advantages:**
 - Dedicated disk IO threads improve work distribution and storage utilization.
 - Minimizes latency and prevents guest system hangs during intense I/O activities by ensuring that neither the main thread nor any virtual CPU thread gets blocked by disk I/O.

Cache Settings

- **Option:** None
- **Benefits:**
 - This setting ensures that write operations are considered complete only when data reaches the physical storage's write queue.
 - It effectively balances safety and performance by bypassing the host's page cache.

AIO Type

- **Selection:** Native
- **Rationale:** Benchmarks indicate that the "native" option leverages "io_uring" effectively, particularly in high-load scenarios, offering superior performance benefits.

Disk Format

- **Choice:** Raw
- **Performance Impact:**
 - The "raw" format is approximately 10% faster than "qcow2".
 - However, it lacks support for snapshot and backup features, which might be a trade-off for some use cases.

Integrating SupremeRAID™ with Proxmox VE

- 1. Installing Proxmox VE:** The process begins with installing the Proxmox Virtual Environment (VE) operating system on your hardware. Keep in mind that SupremeRAID™ cannot be used as a boot solution. Therefore, customers may have to define a boot redundancy that allows them to stay online. Because the boot partition normally does not contain vital information, Graid Technology suggests one of the solutions with RAID 1 using ZFS.
- 2. Install SupremeRAID™ Driver:** After installing Proxmox VE, the next step is to install the SupremeRAID™ driver, which is crucial for the SupremeRAID™ card to be recognized and function correctly within the Proxmox environment.
- 3. Post-Installation Setup:** This step involves creating a Virtual Disk (VD) on the SupremeRAID™ card and setting up the /etc/fstab to auto-mount the drives. SupremeRAID™ will recognize the device as (e.g., /dev/gdgXnY) by the system.
- 4. Setup Proxmox Datastore:** Then set up the datastore in Proxmox using the locate storage created by SupremeRAID™.
- 5. Create Virtual Machines (VMs):** Once the datastore is configured and added to Proxmox, you can proceed to create and manage VMs using the storage space provided by the SupremeRAID™ VD.

Install Proxmox VE

Download Proxmox VE here:

<https://www.proxmox.com/en/downloads/proxmox-virtual-environment/iso>

Note: Ensure you enable virtualization features (Intel VT/AMD-V) in your bios.

Install SupremeRAID™ Drivers

Download the latest version of the SupremeRAID™ driver and pre-installer here:

[SupremeRAID™ Driver Download Page](#)

- SupremeRAID™ pre-installer Package

- Driver installer packages: SR-1010, SR-1000, SR-1001

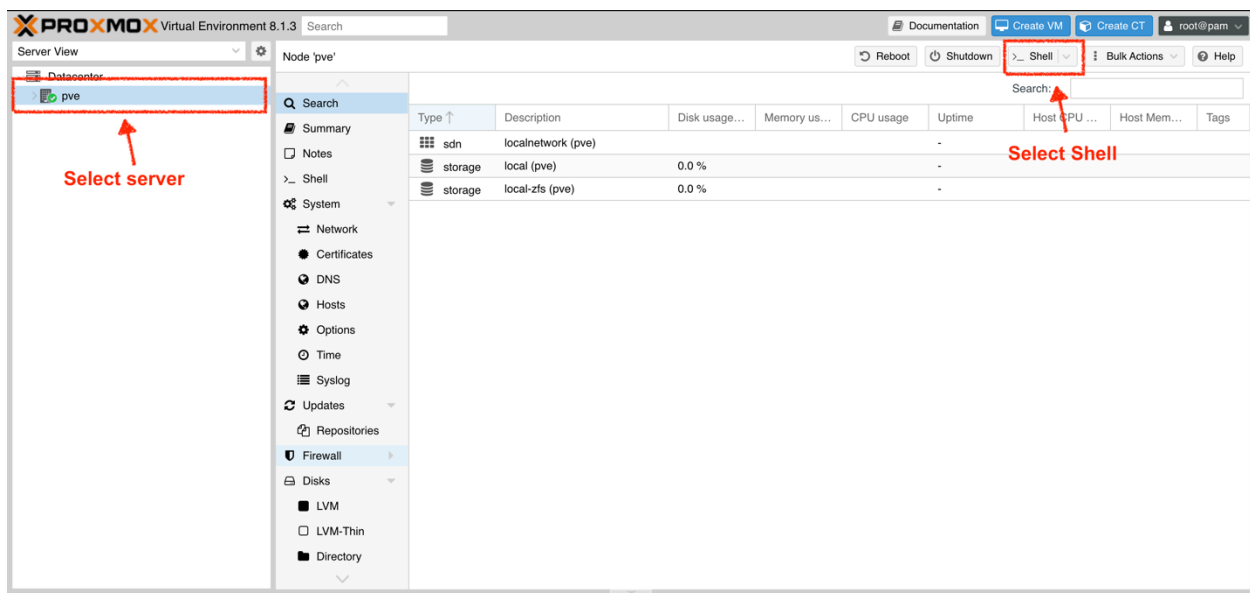
Driver Installation Instructions

Connect to the server via SSH or open a shell on Proxmox.

If using ssh:

```
1 ssh root@[Server IP]
```

Proxmox has a shell option; once you log in, choose a node, then select shell.



Download Pre-Installer

Note: Always choose the most recent version of the driver package for download. This guidance is tailored to the version available as of March 20, 2024, so verifying that you are obtaining the latest release to ensure compatibility and access to the newest features is essential.

```
1 wget https://download.graidtech.com/driver/pre-install/graid-sr-pre-installer-1.5.0-104-x86_64.run
```

The pre-installer will install all necessary dependency packages; carefully follow the instructions for a successful setup.

```
1 $ sudo chmod +x graid-sr-pre-installer-1.5.0-*.run
2 $ sudo ./graid-sr-pre-installer-1.5.0-*.run
```

Download the Latest Driver Package

Note: Always choose the most recent version of the driver package for download. This guidance is tailored to the version available as of March 20, 2024, so verifying that you are obtaining the latest release to ensure compatibility and access to the newest features is essential.

Below, you can choose which pkg is appropriate for your configuration.

```
1 #SR-1010
2 wget https://download.graidtech.com/driver/sr/linux/1.5.0/release/graid-sr-installer-1.5.0-010-700-128.run
3 #SR-1000
4 wget https://download.graidtech.com/driver/sr/linux/1.5.0/release/graid-sr-installer-1.5.0-000-700-128.run
5
6 #SR-1001
7 wget https://download.graidtech.com/driver/sr/linux/1.5.0/release/graid-sr-installer-1.5.0-001-700-128.run
```

Execute the appropriate installer to install the SupremeRAID™ driver. Ensure you follow the provided instructions meticulously for a successful and efficient set-up.

```
1 $ sudo chmod +x graid-sr-installer-1.5.0-*-x86_64.run
2 $ sudo ./graid-sr-installer-1.5.0-*-x86_64.run
```

Post-Installation Setup

This assumes to use a user, but if the root is used, then no sudo is needed. The entire screen sequence of the commands is in one screen below.

SupremeRAID™ Configuration

Activate SupremeRAID™

```
sudo graidctl apply license <license key>
Setup SupremeRAID™ RAID (use RAID5 as example)
Identify how many NVMe_Drives are available
```

```
sudo graidctl list nvme_drive
```

Create Physical Drive (pd)

```
sudo graidctl create physical_drive /dev/nvme0-7
```

Create Drive Group (dg) and build a RAID 5. The output shows 7 NVMe drives available on this example.

```
sudo graidctl create drive_group RAID5 0-7
```

Create Virtual Drive (vd)

```
sudo graidctl create virtual_drive 0
```

```
[graid@graid-demo ~]$ graidctl apply license XXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX
✓Apply license successfully.
```

(Apply license) →

```
[graid@graid-demo ~]$ sudo graidctl list nvme_drive
✓List nvme drive successfully.
```

(Step i) →

DEVICE PATH (7)	MODEL	NQN/WWID	NSID	CAPACITY	NUMA NODE	ADDRESS
/dev/nvme0n1	QEMU NVMe Ctrl	nqn.2019-08.org.qemu:NVME0004	1	1.0 GiB	0	0000:05:00.0
/dev/nvme1n1	QEMU NVMe Ctrl	nqn.2019-08.org.qemu:NVME0005	1	1.0 GiB	0	0000:04:00.0
/dev/nvme2n1	QEMU NVMe Ctrl	nqn.2019-08.org.qemu:NVME0001	1	1.0 GiB	0	0000:08:00.0
/dev/nvme3n1	QEMU NVMe Ctrl	nqn.2019-08.org.qemu:NVME0003	1	1.0 GiB	0	0000:06:00.0
/dev/nvme4n1	QEMU NVMe Ctrl	nqn.2019-08.org.qemu:NVME0002	1	1.0 GiB	0	0000:07:00.0
/dev/nvme5n1	QEMU NVMe Ctrl	nqn.2019-08.org.qemu:NVME0005	1	1.0 GiB	0	0000:09:00.0
/dev/nvme6n1	QEMU NVMe Ctrl	nqn.2019-08.org.qemu:NVME0006	1	1.0 GiB	0	0000:10:00.0
/dev/nvme7n1	QEMU NVMe Ctrl	nqn.2019-08.org.qemu:NVME0007	1	1.0 GiB	0	0000:11:00.0

```
[graid@graid-demo ~]$ sudo graidctl create physical_drive /dev/nvme0-7
✓Create physical drive successfully.
```

(Step ii) →

```
✓Create physical drive PD0 (/dev/nvme0: nqn.2019-08.org.qemu:NVME0004) successfully.
✓Create physical drive PD1 (/dev/nvme1: nqn.2019-08.org.qemu:NVME0005) successfully.
✓Create physical drive PD2 (/dev/nvme2: nqn.2019-08.org.qemu:NVME0001) successfully.
✓Create physical drive PD3 (/dev/nvme3: nqn.2019-08.org.qemu:NVME0003) successfully.
✓Create physical drive PD4 (/dev/nvme4: nqn.2019-08.org.qemu:NVME0002) successfully.
✓Create physical drive PD5 (/dev/nvme4: nqn.2019-08.org.qemu:NVME0005) successfully.
✓Create physical drive PD6 (/dev/nvme4: nqn.2019-08.org.qemu:NVME0006) successfully.
✓Create physical drive PD7 (/dev/nvme4: nqn.2019-08.org.qemu:NVME0007) successfully.
```

(List physical drive) →

```
[graid@graid-demo ~]$ sudo graidctl list physical_drive
✓List physical drive successfully.
```

PD ID (7)	DG ID	DEVICE PATH	NQN/WWID	MODEL	CAPACITY	SLOT ID	NUMA NODE	STATE
0	N/A	/dev/gpd0	nqn.2019-08.org.qemu:NVME0004	QEMU NVMe Ctrl	960 MiB	N/A	0	UNCONFIGURED_GOOD
1	N/A	/dev/gpd1	nqn.2019-08.org.qemu:NVME0005	QEMU NVMe Ctrl	960 MiB	N/A	0	UNCONFIGURED_GOOD
2	N/A	/dev/gpd2	nqn.2019-08.org.qemu:NVME0001	QEMU NVMe Ctrl	960 MiB	N/A	0	UNCONFIGURED_GOOD
3	N/A	/dev/gpd3	nqn.2019-08.org.qemu:NVME0003	QEMU NVMe Ctrl	960 MiB	N/A	0	UNCONFIGURED_GOOD
4	N/A	/dev/gpd4	nqn.2019-08.org.qemu:NVME0002	QEMU NVMe Ctrl	960 MiB	N/A	0	UNCONFIGURED_GOOD
5	N/A	/dev/gpd5	nqn.2019-08.org.qemu:NVME0005	QEMU NVMe Ctrl	960 MiB	N/A	0	UNCONFIGURED_GOOD
6	N/A	/dev/gpd6	nqn.2019-08.org.qemu:NVME0006	QEMU NVMe Ctrl	960 MiB	N/A	0	UNCONFIGURED_GOOD
7	N/A	/dev/gpd7	nqn.2019-08.org.qemu:NVME0007	QEMU NVMe Ctrl	960 MiB	N/A	0	UNCONFIGURED_GOOD

```
[graid@graid-demo ~]$ sudo graidctl create drive_group RAID5 0-7
✓Create drive group successfully.
```

(Step iii) →

```
✓Create drive group DG0 successfully.
```

(List drive group) →

```
[graid@graid-demo ~]$ sudo graidctl list drive_group
✓List drive group successfully.
```

DG ID	MODE	VD NUM	CAPACITY	FREE	USED	CONTROLLER	STATE
0	RAID5	0	6.6 GiB	6.6 GiB	0 B	running: 0 prefer: 0	OPTIMAL

(Step iv) →

```
[graid@graid-demo ~]$ sudo graidctl create virtual_drive 0
✓Create virtual drive successfully.
```

(List virtual drive) →

```
✓Create virtual drive DG0/VD0 successfully.
```

```
[graid@graid-demo ~]$ sudo graidctl list virtual_drive
✓List virtual drive successfully.
```

VD ID	DG ID	SIZE	DEVICE PATH	STATE	EXPORTED
0	0	6.6 GiB	/dev/gdg0n1	OPTIMAL	No

Filesystem Setup

In this document, we have selected "Directory" as the datastore type for Proxmox.

Using Ext4 file system, for example, you can choose any common filesystem you would like to use.

Note: Choose the file system support list by Proxmox; please reference this link:

[Proxmox-Storage](#)

Check the SupremeRAID™ device/s to ensure the Proxmox server can use them.

```
$ lsblk
```

Output:

```
graid@graid-demo:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:16   1 894.3G  0 disk
├─sda1       8:17   1 1007K  0 part
├─sda2       8:18   1    1G  0 part /boot/efi
└─sda3       8:19   1 893.3G  0 part
   └─pve-swap 252:0    0    8G  0 lvm  [SWAP]
      └─pve-root 252:2    0   96G  0 lvm  /
         └─pve-data_tmeta 252:5    0    7.7G  0 lvm
            └─pve-data 252:7    0 757.8G  0 lvm
               └─pve-data_tdata 252:6    0 757.8G  0 lvm
                  └─pve-data 252:7    0 757.8G  0 lvm
sdc          8:32   1    0B  0 disk
sr0         11:0    1 1024M  0 rom
gdg0n1      259:1    0 20.4T  0 disk
```

Create a new directory to use as a mounting point:

```
$ mkdir /mnt/graid
```

Format the SupremeRAID™ device with ext4 file system:

```
$ mkfs.ext4 /dev/gdg0n1
Mount the SupremeRAID™ device on the /mnt/graid with option -o
as showed below. Restart the daemon-reload and execute the mount
to assure it's all correct within the /etc/fstab.
$ mount -o noatime,nodiratime /dev/gdg0n1 /mnt/graid
```

Output:

```
graid@graid-demo:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:16   1 894.3G  0 disk
├─sda1       8:17   1 1007K  0 part
├─sda2       8:18   1    1G  0 part /boot/efi
└─sda3       8:19   1 893.3G  0 part
   ├─pve-swap 252:0    0    8G  0 lvm  [SWAP]
   ├─pve-root 252:2    0   96G  0 lvm  /
   ├─pve-data_tmeta 252:5    0    7.7G  0 lvm
   ├─pve-data 252:7    0 757.8G  0 lvm
   ├─pve-data_tdata 252:6    0 757.8G  0 lvm
   └─pve-data 252:7    0 757.8G  0 lvm
sdc          8:32   1    0B  0 disk
sr0         11:0    1 1024M  0 rom
gdg0n1      259:1    0 20.4T  0 disk /mnt/graid
```

Add the following single line to allow auto mount function. This is the single line in the in /etc/fstab file that ensures the device remounts the drive once the system reboots. Create a copy of the fstab file to be sure all is done correctly before the mount. Restore if something is not working before the reboot.

```
$ cp /etc/fstab /tmp/
$ echo "/dev/gdg0n1 /mnt/graid ext4 x-
systemd.requires=graid.service,nofail,noatime,nodiratime 0 0" >>
/etc/fstab
```

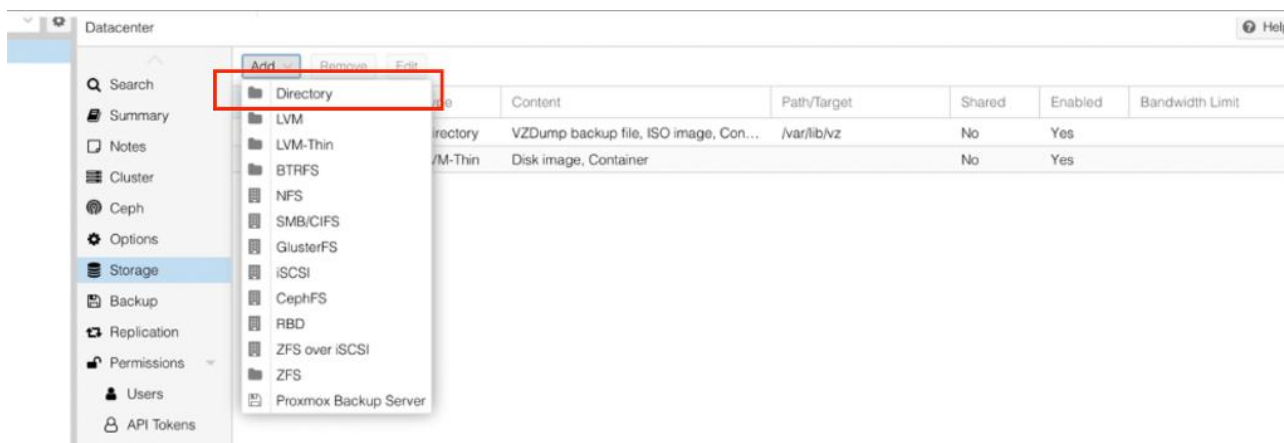
In the fstab's first field (fs_spec), you can use the device path or udev. To query udev, use "ls -l /dev/disk/by-id/ | grep -i gdg0n1".

```
graid@graid-demo:~$ cat /etc/fstab
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/pve/root / ext4 errors=remount-ro 0 1
UUID=AD90... /boot/efi vfat defaults 0 1
/dev/pve/swap none swap sw 0 0
pve /proc/proc defaults 0 0
/dev/gdg0n1 /mnt/graid ext4 x-systemd.requires=graid.service,x-systemd.automount,nofail,noatime,nodiratime,nobarrier 0 0
```

The commands executed above will look like the following (sudo is not required as you are root):

```
graid@graid-demo:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:16   1 894.3G  0 disk
├─sda1       8:17   1 1007K  0 part
├─sda2       8:18   1    1G   0 part /boot/efi
└─sda3       8:19   1 893.3G  0 part
   └─pve-swap 252:0    0    8G   0 lvm  [SWAP]
   └─pve-root 252:1    0   96G   0 lvm  /
   └─pve-data_tmeta 252:2    0   7.7G   0 lvm
   └─pve-data 252:7    0 757.8G   0 lvm
   └─pve-data_tdata 252:3    0 757.8G   0 lvm
   └─pve-data 252:7    0 757.8G   0 lvm
sdc          8:32   1    0B   0 disk
sr0         11:0    1 1024M   0 rom
gdg0n1      259:18   0   4.7G   0 disk
graid@graid-demo:~$ sudo mkdir -p /mnt/graid
graid@graid-demo:~$ sudo mkfs.ext4 /dev/gdg0n1
graid@graid-demo:~$ echo "/dev/gdg0n1 /mnt/graid/ ext4 x-systemd.requires=graid.service,x-systemd.automount,nofail,noatime,nodiratime,nobarrier 0 0" |sudo tee
-a /etc/fstab
graid@graid-demo:~$ cat /etc/fstab
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/pve/root / ext4 errors=remount-ro 0 1
UUID=AD99-ABA9 /boot/efi vfat defaults 0 1
/dev/pve/swap none swap sw 0 0
proc /proc proc defaults 0 0
/dev/gdg0n1 /mnt/graid/ ext4 x-systemd.requires=graid.service,x-systemd.automount,nofail,noatime,nodiratime,nobarrier 0 0
graid@graid-demo:~$ sudo systemctl daemon-reload
graid@graid-demo:~$ sudo mount -a
graid@graid-demo:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:16   1 894.3G  0 disk
├─sda1       8:17   1 1007K  0 part
├─sda2       8:18   1    1G   0 part /boot/efi
└─sda3       8:19   1 893.3G  0 part
   └─pve-swap 252:0    0    8G   0 lvm  [SWAP]
   └─pve-root 252:1    0   96G   0 lvm  /
   └─pve-data_tmeta 252:2    0   7.7G   0 lvm
   └─pve-data 252:7    0 757.8G   0 lvm
   └─pve-data_tdata 252:3    0 757.8G   0 lvm
   └─pve-data 252:7    0 757.8G   0 lvm
sdc          8:32   1    0B   0 disk
sr0         11:0    1 1024M   0 rom
gdg0n1      259:18   0   4.7G   0 disk /mnt/graid
```

Once logged into Proxmox (using the web interface), navigate to the "Datacenter" tab and then select "Storage". Click on the "Add" button and select "Directory" from the dropdown menu. There are several options in this section, including LVM thin provisioning, volume group selection, and thin pool configuration.



In the "Add Directory" window, input a unique ID for your new storage. Then, in the "Directory" field, enter the mount point you previously established (for example, /mnt/graid). Next, select the types of content you wish to store, such as "Disk image," "Container template," or "Backup" from the available options. Finally, click "Add" to confirm and save your new storage configuration.

The screenshot shows the 'Add: Directory' configuration window. It has two tabs: 'General' (selected) and 'Backup Retention'. The 'General' tab contains the following fields:

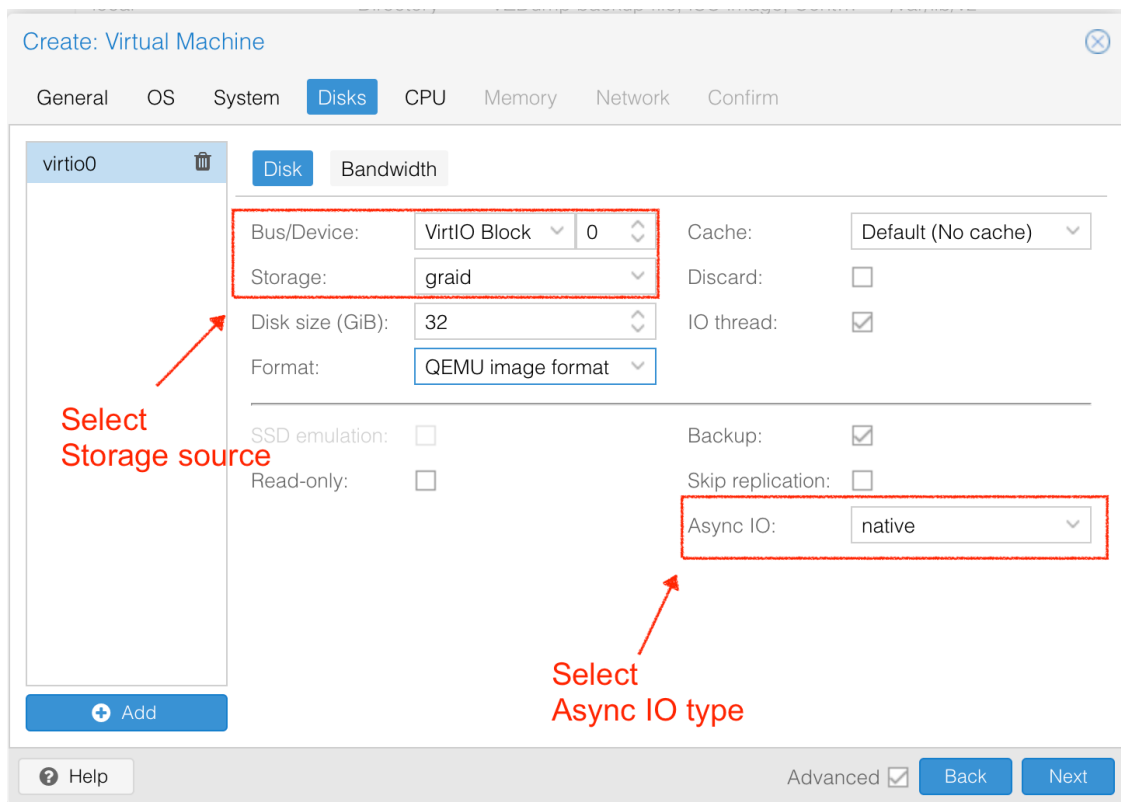
- ID:** A text input field containing 'graid'.
- Nodes:** A dropdown menu set to 'All (No restrictions)'.
- Directory:** A text input field containing '/mnt/graid', highlighted with a red box.
- Content:** A dropdown menu set to 'Disk image'.
- Enable:** A checked checkbox.
- Shared:** An unchecked checkbox.

At the bottom, there is a 'Help' button, a red arrow pointing to the 'Content' dropdown with the text 'fill in the mount point', an 'Advanced' checkbox, and an 'Add' button.

Creation of Virtual Machines (VMs)

After finishing the datastore setup on the storage page, you can create a VM with the SupremeRAID™ device.

1. Click "Create VM"
2. On the Disk page, select "VirtIO Block" and choose the Disk just created



3. Set the Virtual Machine system parameters, then finish.

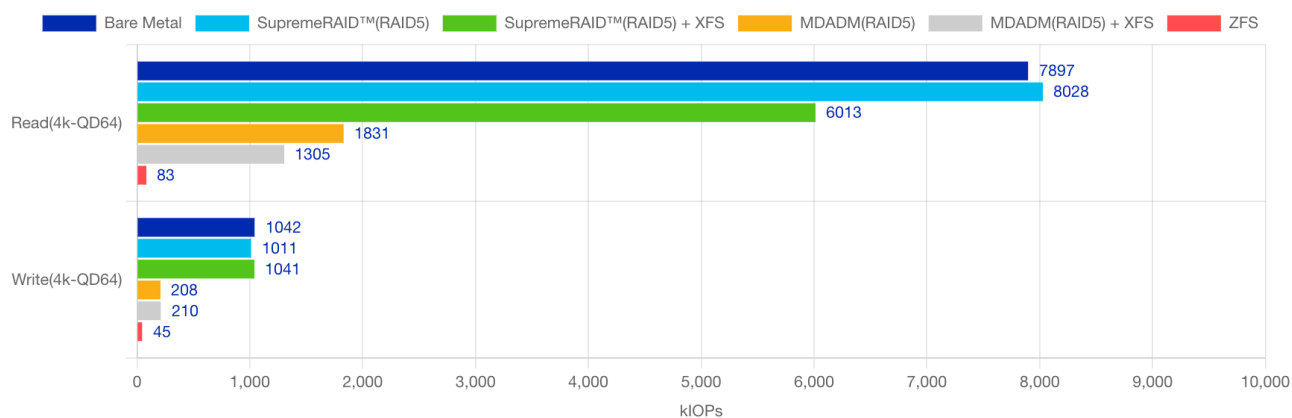
Note: Due to SupremeRAID™'s architecture, which obscures the device paths of the drives it manages, users utilizing the Proxmox management web interface will encounter a specific limitation. After implementing SupremeRAID™ into their system, these drives will no longer appear in the standard Proxmox web UI. However, it's important to note that users can still access the S.M.A.R.T information of these drives via the command line. This can be achieved by executing the command `"sudo smartctl -d nvme -a /dev/gpd<#>"`, allowing for continued monitoring and management of drive health and performance, even with the device paths hidden by SupremeRAID™.

Performance Comparison

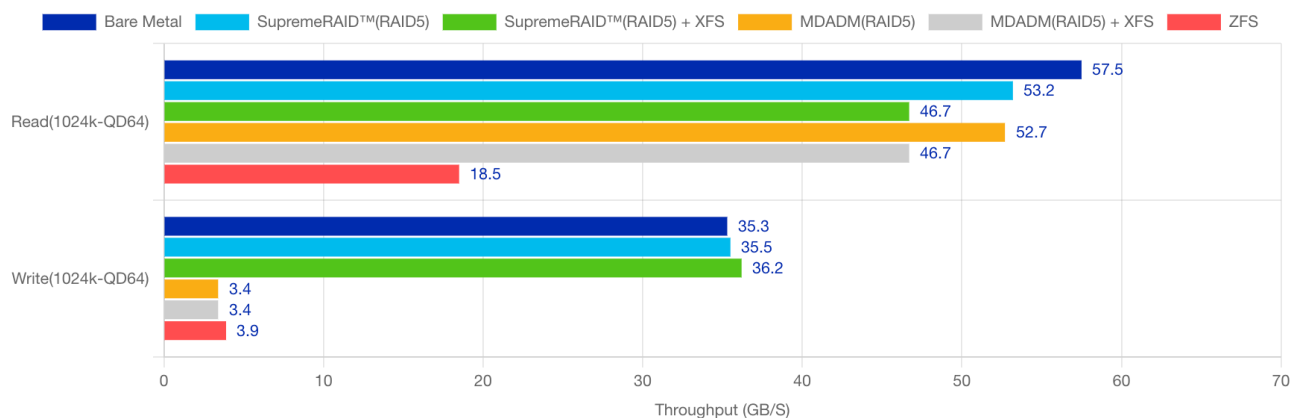
Baseline Benchmarks Among Various RAID Configurations

We conducted timed tests on various configurations, including Bare Metal (without RAID), SupremeRAID™, MDADM, and ZFS, to establish a solid baseline and thoroughly understand their peak performance capabilities. We focused on comparing block device and filesystem performance in each setup, using XFS as our filesystem of choice. Our systematic approach aimed to precisely evaluate and compare the efficiency and throughput of SupremeRAID™, MDADM, and ZFS in a controlled, uniform testing environment.

Aggregate FIO Performance Summary (Random IO) (Bare Metal vs. SupremeRAID™ vs. MDADM vs. ZFS)



Aggregate FIO Performance Summary (Sequential IO) (Bare Metal vs. SupremeRAID™ vs. MDADM vs. ZFS)

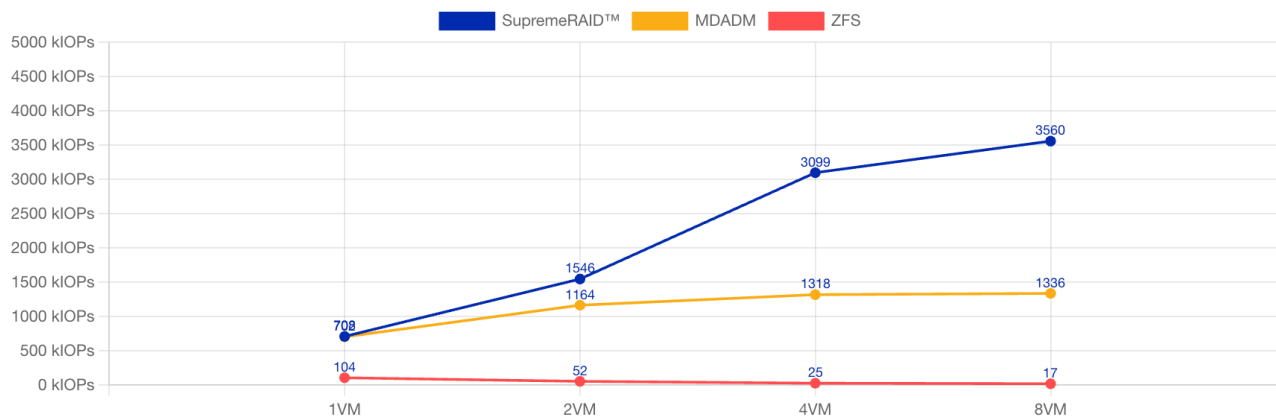


Virtual Machine Performance

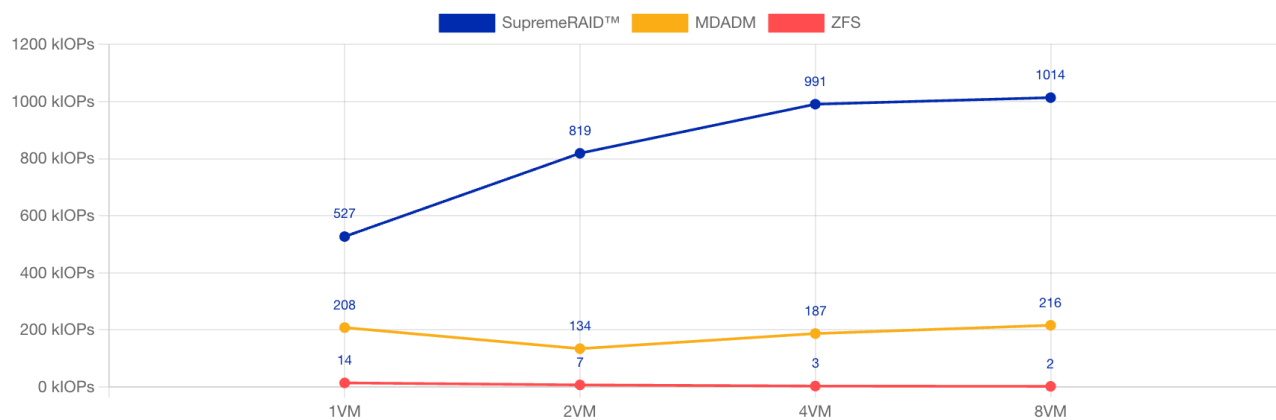
To gauge each solution's performance scalability and workload management capabilities within the Proxmox Virtual Machines environment, Graid Technology initiated a simulation that utilized six disks in each VM for precise performance benchmarking. Furthermore, the evaluation was broadened to encompass a detailed test involving 48 VMs, each equipped with a single disk, to explore the performance dynamics as the Proxmox service scales up.

This methodical approach offered an in-depth and wide-ranging insight into the system's overall performance and efficiency under varied demand levels. The objective of this section is to conduct a comprehensive analysis of the performance of each RAID solution, including SupremeRAID™, MDADM, and ZFS, in a highly virtualized setting.

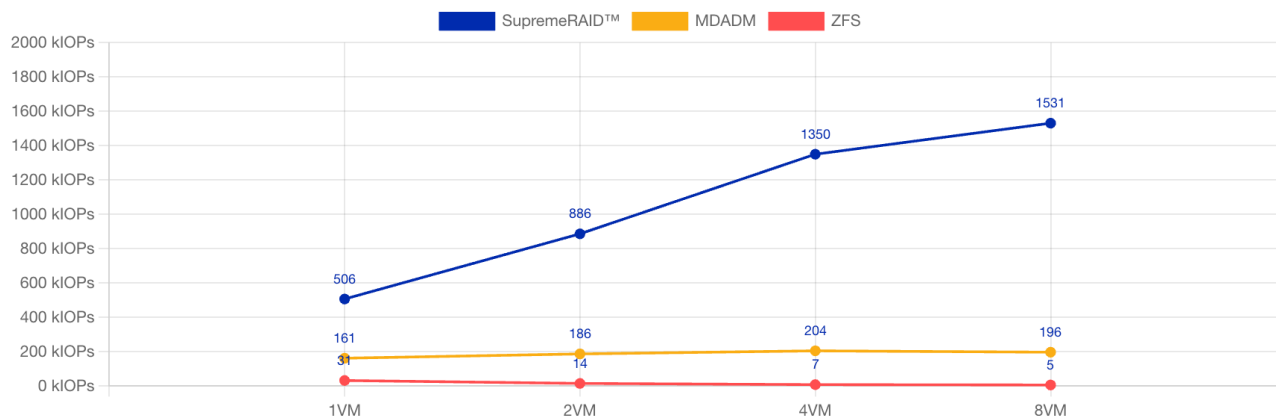
Aggregate FIO Random Read Performance Across Varied VM



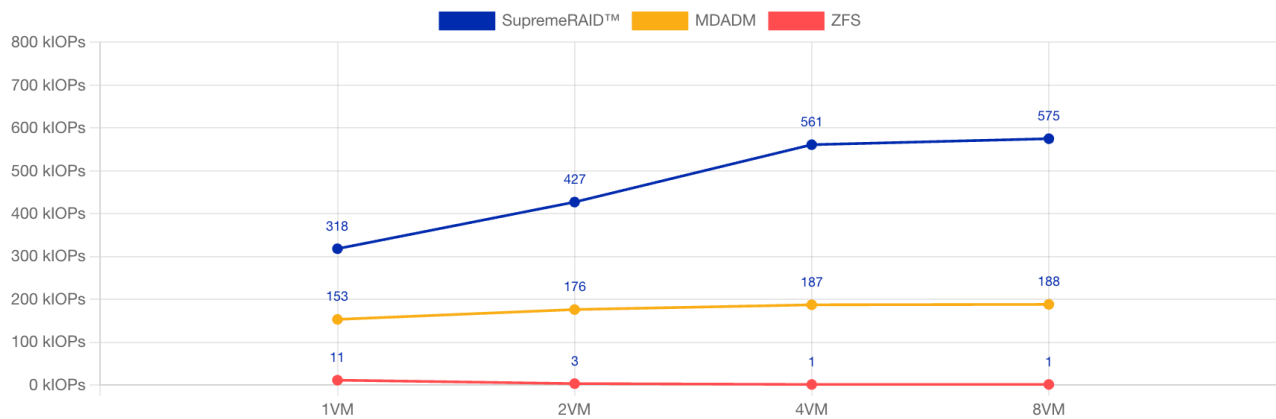
Aggregate FIO Random Write Performance Across Varied VM



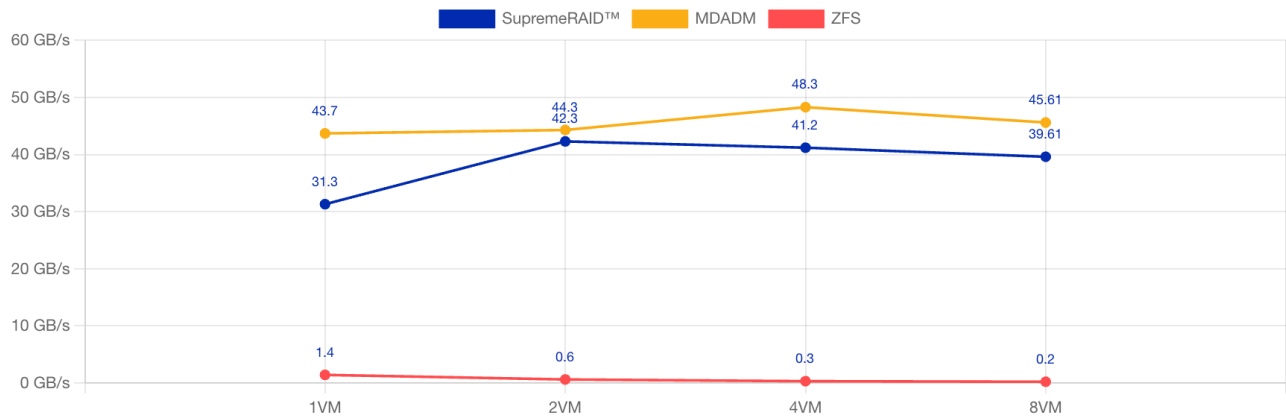
Aggregate FIO Random Read (Rebuilding) Performance Across Varied VM



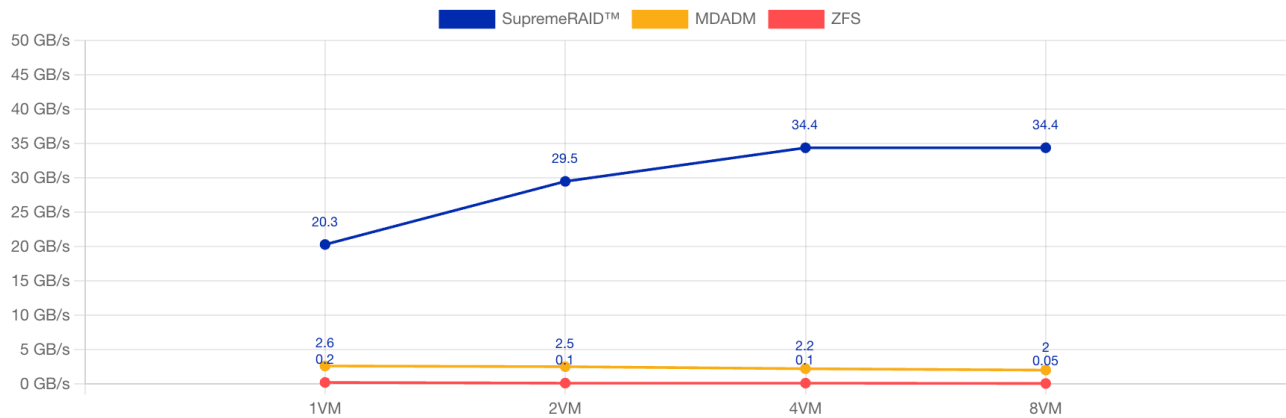
Aggregate FIO Random Write (Rebuilding) Performance Across Varied VM



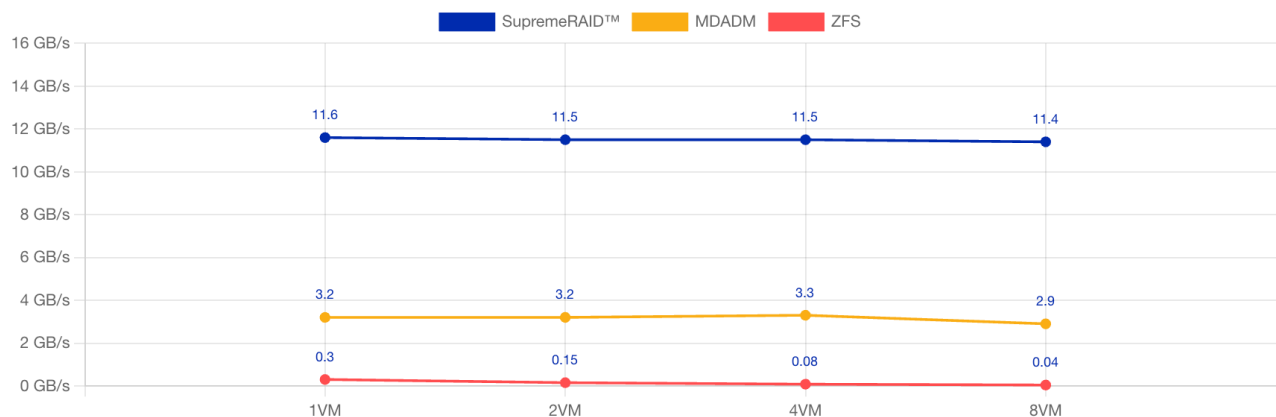
Aggregate FIO Sequential Read Performance Across Varied VM



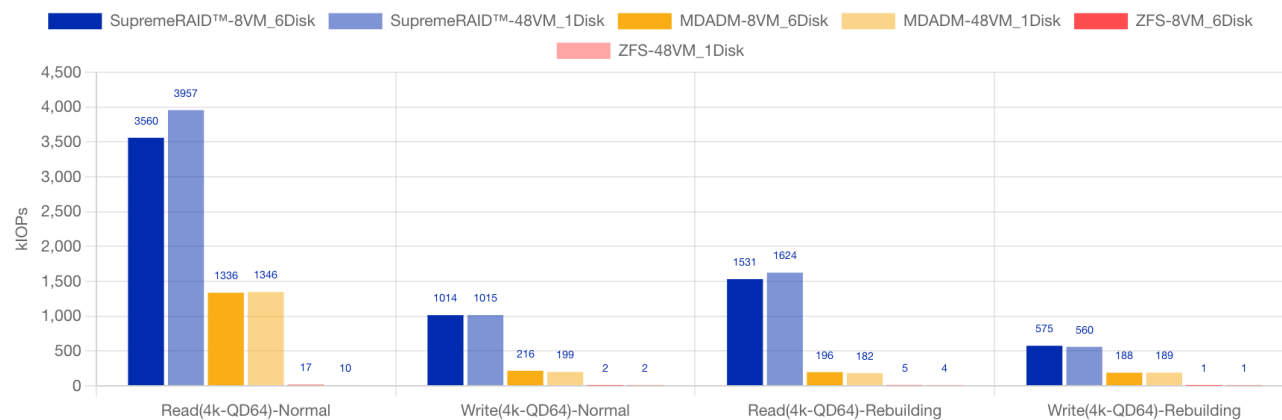
Aggregate FIO Sequential Write Performance Across Varied VM



Aggregate FIO Sequential Read (Rebuilding) Performance Across Varied VM

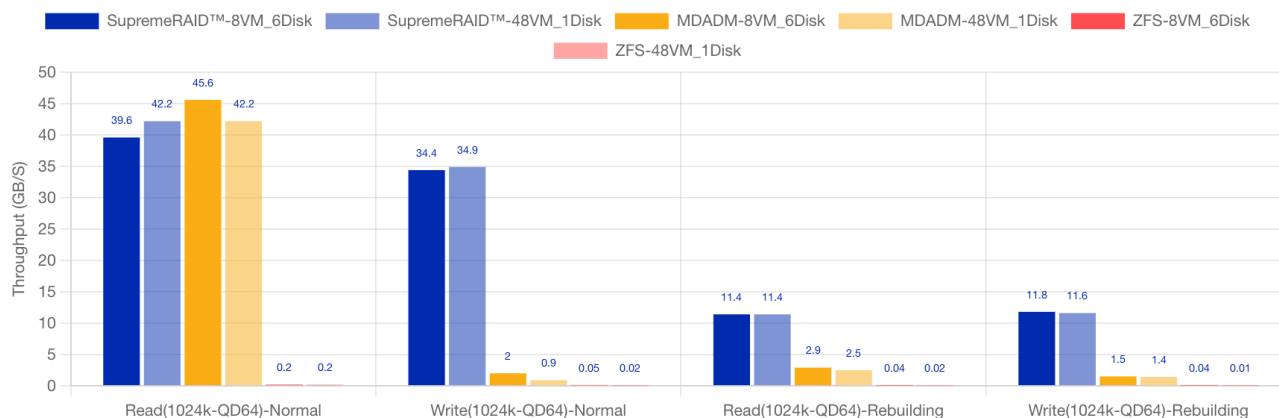


Aggregate FIO Performance Summary (Random IO) 8VM (6 Disk) vs 48VM (1 Disk)



Aggregate FIO Performance Summary (Sequential IO)

8VM (6 Disk) vs 48VM (1 Disk)



Conclusion

Expanding upon the impressive outcomes of these tests, SupremeRAID™ not only excels in delivering consistent performance across multi-VM scenarios but also emerges as a transformative force in the areas of data storage and management. One of SupremeRAID™'s key strengths is its ability to markedly diminish the Total Cost of Ownership (TCO), making it an appealing choice for Cloud Service Providers (CSPs) and IT users aiming to enhance operational efficiency and optimize budget utilization.

Furthermore, the enhanced performance of SupremeRAID™ is particularly noteworthy, especially its capability to uphold performance levels that are 2 to 10 times higher than alternative solutions during critical RAID rebuild phases. This feature holds immense significance in demanding environments where downtime or decreased performance could lead to substantial operational repercussions and underscores the sophisticated efficiency and technological superiority of SupremeRAID™.

Additionally, the ease of maintenance and reduced management effort required to oversee SupremeRAID™ systems offers a compelling advantage. This streamlined approach to maintenance reduces the burden on IT teams and culminates in more steadfast and reliable system performance over time. This reliability is essential for enterprises where data integrity and accessibility are paramount.

Moreover, the adaptable and scalable architecture of SupremeRAID™ ensures its sustainability and future readiness as storage requirements evolve and intensify. This adaptability is indispensable in an era marked by the exponential growth of data generation and storage needs.

In closing, incorporating SupremeRAID™ into IT infrastructures presents a comprehensive enhancement in performance, efficiency, and dependability. Its advanced functionalities and capabilities make it a vital tool for businesses and IT environments requiring high-performance, easy maintenance, and expandable storage solutions. SupremeRAID™ represents an advancement in RAID technology and a significant leap forward in state-of-the-art data management solutions.

Appendix

Test Environment System Information

Hardware Specifications

```
Processor: AMD EPYC 74F3 24-Core Processor @ 3.20GHz x 2
Memory: Samsung M393A8G40AB2-CWE 64GB DDR4 3200Mhz x 32
RAID Controller: SupremeRAID™ SR-1000 SR-BUN-1010-12-FD32 x 1
SSD: Phison EPW0970-3200GB x 8
```

Software Configurations

```
OS: Proxmox 8.1.4
Kernel: 6.5.11-8-pve
SupremeRAID™ Driver Version: 1.5.0-699
mdadm version: v4.2 - 2021-12-30
ZFS version: zfs-2.2.2-pve1
Filesystem: xfs 6.1.0-1
Benchmark Tool: fio-3.36
```

SupremeRAID™ RAID Configuration

PD ID (8)	DG ID	DEVICE PATH	NQN/WWID	MODEL	CAPACITY	SLOT ID	NUMA NODE	STATE
0	0	/dev/gpd0	nqn.2019-12.com.seagate:nvme:nvm-subsystem-sn-7XR002EQ	EPW0970-3200GB	2.9 TiB	7	1	ONLINE
1	0	/dev/gpd1	nqn.2019-12.com.seagate:nvme:nvm-subsystem-sn-7XR002JW	EPW0970-3200GB	2.9 TiB	5	1	ONLINE
2	0	/dev/gpd2	nqn.2019-12.com.seagate:nvme:nvm-subsystem-sn-7XR002GD	EPW0970-3200GB	2.9 TiB	2	1	ONLINE
3	0	/dev/gpd3	nqn.2019-12.com.seagate:nvme:nvm-subsystem-sn-7XR002KA	EPW0970-3200GB	2.9 TiB	8	1	ONLINE
4	0	/dev/gpd4	nqn.2019-12.com.seagate:nvme:nvm-subsystem-sn-7XR002K5	EPW0970-3200GB	2.9 TiB	6	1	ONLINE
5	0	/dev/gpd5	nqn.2019-12.com.seagate:nvme:nvm-subsystem-sn-7XR002KV	EPW0970-3200GB	2.9 TiB	3	1	ONLINE
6	0	/dev/gpd6	nqn.2019-12.com.seagate:nvme:nvm-subsystem-sn-7XR002L6	EPW0970-3200GB	2.9 TiB	9	1	ONLINE
7	0	/dev/gpd7	nqn.2019-12.com.seagate:nvme:nvm-subsystem-sn-7XR002GT	EPW0970-3200GB	2.9 TiB	4	1	ONLINE

DG ID	MODE	VD NUM	CAPACITY	FREE	USED	CONTROLLER	STATE
0	RAID5	1	20 TiB	385 GiB	20 TiB	running: 0 prefer: 0	OPTIMAL

VD ID	DG ID	SIZE	DEVICE PATH	STATE	EXPORTED
0	0	20 TiB	/dev/gdg0n1	OPTIMAL	No

MDADM RAID Configuration and Settings

```

sudo mdadm --create /dev/md0 -c 128K --auto=yes --verbose --
level=5 --assume-clean -n 8 \
/dev/nvme1n1 /dev/nvme2n1 /dev/nvme3n1 /dev/nvme4n1 /dev/nvme5n1
/dev/nvme6n1 /dev/nvme7n1 /dev/nvme8n1
declare -A settings
settings["group_thread_cnt"]=12
settings["sync_speed_min"]=1000000
settings["sync_speed_max"]=2000000
for key in "${!settings[@]}; do
    echo "${settings[$key]}" > "/sys/block/md0/md/$key"
    sleep 5
    cat "/sys/block/md0/md/$key"
done

```

RAID Configuration

```

Personalities : [raid6] [raid5] [raid4] [linear] [multipath]
[raid0] [raid1] [raid10]
md0 : active raid5 nvme8n1[7] nvme7n1[6] nvme6n1[5] nvme5n1[4]
nvme4n1[3] nvme3n1[2] nvme2n1[1] nvme1n1[0]
      21878391808 blocks super 1.2 level 5, 128k chunk,

```

```
algorithm 2 [8/8] [UUUUUUUU]
    bitmap: 15/24 pages [60KB], 65536KB chunk

unused devices:
```

ZFS Configuration

```
sudo zpool create -o ashift=12 zfs-pool -O recordsize=1M -O
compression=lz4 -O dedup=off -O atime=off -O xattr=sa \
-O checksum=off -O logbias=throughput raidz \
/dev/nvme1n1 /dev/nvme2n1 /dev/nvme3n1 /dev/nvme4n1 /dev/nvme5n1
/dev/nvme6n1 /dev/nvme7n1 /dev/nvme8n1
sudo zfs create zfs-pool/raidz
```

RAID Configuration

```
pool: zfs-pool
state: ONLINE
config:
```

NAME	STATE	READ	WRITE	CKSUM
zfs-pool	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
nvme1n1	ONLINE	0	0	0
nvme2n1	ONLINE	0	0	0
nvme3n1	ONLINE	0	0	0
nvme4n1	ONLINE	0	0	0
nvme5n1	ONLINE	0	0	0
nvme6n1	ONLINE	0	0	0
nvme7n1	ONLINE	0	0	0
nvme8n1	ONLINE	0	0	0

```
errors: No known data errors
```

Virtual Machine Configuration

```
args:
boot: order=virtio0;ide2;net0
cores: 6
memory: 8192
meta: creation-qemu=7.2.0,ctime=1703223884
name: testing
net0: virtio=FE:39:33:0C:AA:43,bridge=vmbr0,firewall=1
numa: 0
ostype: l26
scsihw: virtio-scsi-single
smbios1: uuid=7fc04b51-5082-4d2a-821b-9ff40f35ebcc
sockets: 1
virtio0: graid:104/vm-104-disk-
0.raw,aio=native,iothread=1,size=20G
virtio1: graid:104/vm-104-disk-
1.qcow2,aio=native,iothread=1,size=20G
virtio2: graid:104/vm-104-disk-
2.qcow2,aio=native,iothread=1,size=20G
virtio3: graid:104/vm-104-disk-
3.qcow2,aio=native,iothread=1,size=20G
virtio4: graid:104/vm-104-disk-
4.qcow2,aio=native,iothread=1,size=20G
virtio5: graid:104/vm-104-disk-
5.qcow2,aio=native,iothread=1,size=20G
virtio6: graid:104/vm-104-disk-
6.qcow2,aio=native,iothread=1,size=20G
vmgenid: 2e19e789-d5e5-40b4-8729-5f5bafb631cf
```

FIO Script for VM Testing

Random Read

```
[global]
time_based=1
randrepeat=0
ioengine=libaio
direct=1
random_generator=tausworthe64
cpus_allowed_policy=split
```

```
group_reporting=1
norandommap=1
rw=randread
bs=4k
iodepth=64
ramp_time=5
runtime=360

[b]
filename=/dev/vdb
numjobs=1
cpus_allowed=0

[c]
filename=/dev/vdc
numjobs=1
cpus_allowed=1

[d]
filename=/dev/vdd
numjobs=1
cpus_allowed=2

[e]
filename=/dev/vde
numjobs=1
cpus_allowed=3

[f]
filename=/dev/vdf
numjobs=1
cpus_allowed=4

[g]
filename=/dev/vdg
numjobs=1
cpus_allowed=5

Sequential Read
[global]
time_based=1
randrepeat=0
ioengine=libaio
direct=1
```

```
random_generator=tausworthe64
cpus_allowed_policy=split
group_reporting=1
norandommap=1
rw=read
bs=1m
iodepth=64
ramp_time=5
runtime=360
```

```
[b]
filename=/dev/vdb
numjobs=1
cpus_allowed=0
```

```
[c]
filename=/dev/vdc
numjobs=1
cpus_allowed=1
```

```
[d]
filename=/dev/vdd
numjobs=1
cpus_allowed=2
```

```
[e]
filename=/dev/vde
numjobs=1
cpus_allowed=3
```

```
[f]
filename=/dev/vdf
numjobs=1
cpus_allowed=4
```

```
[g]
filename=/dev/vdg
numjobs=1
cpus_allowed=5
Sequential Write
```

```
[global]
time_based=1
randrepeat=0
ioengine=libaio
```

```
direct=1
random_generator=tausworthe64
cpus_allowed_policy=split
group_reporting=1
norandommap=1
rw=write
bs=1m
iodepth=64
ramp_time=5
runtime=360
```

```
[b]
filename=/dev/vdb
numjobs=1
cpus_allowed=0
```

```
[c]
filename=/dev/vdc
numjobs=1
cpus_allowed=1
```

```
[d]
filename=/dev/vdd
numjobs=1
cpus_allowed=2
```

```
[e]
filename=/dev/vde
numjobs=1
cpus_allowed=3
```

```
[f]
filename=/dev/vdf
numjobs=1
cpus_allowed=4
```

```
[g]
filename=/dev/vdg
numjobs=1
cpus_allowed=5
```

Additional Resources

[Blockbridge-OPTIMIZING PROXMOX: IOTHEADS, AIO, & IO URING](#)

[Intel-KVM/Qemu Virtualization Tuning Guide on 3rd Generation](#)

[Intel® Xeon® Scalable Processors Based Platform](#)

[PVE-Main Page](#)

Please note that while every effort has been made to ensure the accuracy of this guide, the rapidly evolving nature of open-source software means that some elements may change over time. Always back up your data and configurations before making significant changes to your system. For additional SupremeRAID™ support or questions, contact the Graid support team at info@graidtech.com. For more detailed Proxmox settings, contact the Proxmox support team at <https://www.proxmox.com/en/services/support>.

Note: Proxmox VE highly recommends subscribing to their services for production environments to ensure optimal support and access to stable software updates. If your environment subscription is inactive or you're considering a non-subscription approach, detailed guidance and alternatives are available on the Proxmox Package Repositories webpage. For comprehensive information about non-subscription options and how to best utilize them, please visit [Proxmox \[Package Repositories\]](#)

About Graid Technology

Graid Technology, creators of SupremeRAID™ next-generation GPU-based RAID, is led by a team of experts in the storage industry and is headquartered in Silicon Valley, California, with an R&D center in Taipei, Taiwan. Winners of the esteemed Tech Trailblazers Storage Award for 2023 and the 2023 Golden Award at the Asia Pacific ICT Alliance, SupremeRAID™ performance is breaking world records as the first NVMe and NVMeoF RAID card to unlock the full potential of your SSD performance: a single SupremeRAID™ card delivers 28 million IOPS and 260GB/s of throughput. For more information on Graid Technology, visit graidtech.com or connect with us on [Twitter](#) or [LinkedIn](#).