Article
Murray Oldfield · May 22, 2018 9m read

LVM PE Striping to maximize Hyper-Converged storage throughput

This post provides useful links and an overview of best practice configuration for low latency storage IO by creating LVM Physical Extent (PE) stripes for database disks on InterSystems Data Platforms; InterSystems IRIS, Caché, and Ensemble.

Consistent low latency storage is key to getting the best database application performance. For applications running on Linux, Logical Volume Manager (LVM) is often used for database disks, for example, because of the ability to grow volumes and filesystems or create snapshots for online backups. For database applications, the parallelism of writes using LVM PE striped logical volumes can also help increase performance for large sequential reads and writes by improving the efficiency of the data I/O.

This post has a focus on using LVM PE stripes with HCI and was also prompted by publication of the white paper Software Defined Data Centers (SDDC) and Hyper-Converged Infrastructure (HCI) – Important Considerations for InterSystems Clients here on the community. The white paper recommended " use of LVM PE striping with Linux virtual machines, which spreads IO across multiple disk groups " and to " Use of the Async IO with the rtkaio library for all databases and write image journal (WIJ) files with Linux virtual machines " . This post provides some context to those requirements and examples.

NOTE:

> Currently there are multiple Hyper-Converged, Converged and Software Defined vendor platforms, rather than provide detailed instructions for each I have used the configuration of **InterSystems IRIS or Caché on Red Hat Enterprise Linux (RHEL) 7.4 running on VMware ESXi and vSAN** as the example in this post. However, the basic process is similar for other solutions, especially at the InterSystems IRIS or Caché and operating system level. If you are unsure how to translate these instructions to other platforms, please contact the respective vendor ' s support for their best practice. InterSystems technology experts can also advise directly with customers and vendors or through the community.

It ' s also worth noting that the guidance on LVM PE striping in this post can be applied to both HCI and " traditional " storage.

Do you have to use LVM striping?

For traditional storage such as disk arrays, the short answer is no. It is not mandatory especially with modern all-flash storage arrays to run LVM striped volumes for your database disks if your performance is ok now and you have no requirement for LVM then you do not have to change.

However, as stated above; LVM stripes are recommended for database disks on Hyper-Converged and storage solutions like Nutanix and VMware vSAN to allow for more host nodes and disk groups to be used in IO operations.

Why use LVM Stripes for Data Platforms?

LVM stripes are especially recommended for the database disks on HCI to mitigate the performance overhead of some features of the architecture, such as to lessen the impact of the Write Daemon (WD) on database writes and journal writes. Using an LVM stripe spreads the database write burst across more disk devices and multiple disk groups. In addition, this post also shows how to increase the parallelism of the large IO size Write Image Journal

(WIJ) which lessens the impact on latency for other IOs.

> Note: In this post when I say " disk " I mean NVMe, Optane, SATA or SAS SSD, or any other flash storage devices.

vSAN storage architecture Overview

HCI storage, for example, when running ESXi on vSAN, uses two disk tiers; a cache tier and a capacity tier. For an all-flash architecture **(you must use all flash - do not use spinning disks!)** all writes go to the cache tier with data then ultimately destaged to the capacity tier. Reads come from the capacity tier (or possibly from cache on the Cache tier). Each host in the HCI cluster can have one or more disk groups. Where there are disk groups, for example with vSAN, each disk group is made up of a cache disk and multiple capacity disks. For example, the cache disk is a single NVMe disk and the capacity disks are three or more write intensive SAS SSD disks.

For further details on HCI, including vSAN disk groups, see the community post Hyper-Converged Infrastructure (HCI) on the community or contact your HCI vendor.

LVM Striped logical volumes overview

A good overview of Linux LVM is available on the Red Hat Support web site and also other places, for example, this tutorial for system administrators is very good.

Data Platforms storage IO

It is important you understand the types of IO generated by InterSystems Data Platforms. An overview of storage IO patterns is available on the community.

Process to create LVM PE stripe

Prerequisites and procedure

Before we dive into the process you should also remember that other variables can come into play that impact storage performance. Simply creating an LVM stripe is not a guarantee of optimal performance, you will also have to consider the storage type, and the whole IO path, including IO queues and queue depth.

This example is for VMware, the InterSystems IRIS VMware best practice guide should also be read and recommendations applied. Especially considerations for storage such as separation of storage IO types across PVSCSI controllers.

Overview

The following example is for best practice using InterSystems IRIS or Caché on Red Hat Enterprise Linux (RHEL) 7.4 running on VMware ESXi and vSAN 6.7.

The following steps are outlined below;

1. ESXi configuration
2. RHEL configuration
3. Caché / InterSystems IRIS configuration

---

1. ESXi Configuration

---

a) Create VMDK disks

You must create disks as per the InterSystems IRIS VMware best practice guide; Databases, journal and WIJ are on separate PVSCI devices.

Create the number of VMDKs depending on your sizing requirements. In this example, the database file system will be made up of four 255GB VMDK disks that will be striped together to create a 900GB logical disk for the database filesystem.

Steps:

1. Power off the VM before adding the VMDKs,
2. In the vCenter console create multiple disks (VMDKs) at 255GB each, all disks in a single LVM stripe must be associated with the same PVSCSI controller.
3. Power on the VM. During power on the new disks will be created at the operating system, for example /dev/sdi etc.

   Why create multiple 255 GB VMDKs? in vSAN storage components are created in chunks of 256GB, keeping the VMDK size just under 256 GB we are trying force the components to be on separate disk groups. Enforcing another level of striping (This has been the case in my testing, but I cannot guarantee that vSAN will actually do this).

Note: During creation vSAN spreads the disk components across all hosts and across disk groups for availability. For example in the case of Failures To Tolerate (FTT) set to 2 there are three copies of each disk component plus two small witness components all on separate hosts. In the event of a disk group, host or network failure the application continues without data loss using the remaining disk component. It is possible to overthink this process! With HCI solutions like vSAN, there is no control over what physical disk the components that make up the VMDKs will reside on at any point in time. In fact, due to maintenance, resynchronisation, or rebuilds, the VMDKs could move to different disk groups or hosts over time. This is OK.

---

2. RHEL Configuration

a) Confirm RHEL IO scheduler is NOOP for each of the disk devices.

The best practice is to use the ESXi kernel ' s scheduler. For more information on setting the scheduler see the Red Hat knowledge base article. We recommend using the option to set for all devices at boot time. To validate you have set the scheduler correctly you can display the current setting for a disk device, for example in this case /dev/sdi as follows;

```
[root@db1 ~]# cat /sys/block/sdi/queue/scheduler
[noop] deadline cfq
```

You can see noop is enabled because it is highlighted between the square brackets.

b) Create striped LVM and XFS file system

We are now ready to create the LVM stripe and database filesystem in RHEL. Following is an example of the steps involved, note the made-up names vgmydb, lvmydb01, and path /mydb/db would be substituted for your environment.

Steps

1. Create the Volume Group with new disk devices using the vgcreate command.

```
vgcreate -s 4M <vg name> <list of all disks just created>
```

For example, if disks /dev/sdh, /dev/sdi, /dev/sdj and /dev/sdk were created:

```
vgcreate -s 4M vgmydb /dev/sd[h-k]
```

2. Create the striped Logical Volume using the lvcreate command. A minimum of four disks is recommended. Start with a 4MB stripe, however with very large logical volumes you may be prompted for a larger size such as 16M.

```
lvcreate -n <lv name> -L <size of LV> -i <number of disks in volume group> -I 4MB <vg
 name>
```

For example to create the 900GB disk with 4 stripes and stripe size of 4 MB :

```
lvcreate -n lvmydb01 -L 900G -i 4 -I 4M vgmydb
```

3. Create the database file system using the mkfs command.

```
mkfs.xfs -K <logical volume device>
```

For example:

```
mkfs.xfs -K /dev/vgmydb/lvmydb01
```

4. Create the file system mount point, for example:

```
mkdir /mydb/db
```

5. Edit /etc/fstab with following mount entries and mount the file system. For example:

```
/dev/mapper/vgmydb-lvmydb01 /mydb/db xfs defaults 0 0
```

6. Mount the new filesystem.

```
mount /mydb/db
```

3. Caché/InterSystems IRIS configuration

In this section we will configure:

- Asynchronous and direct IO for optimal write performance on the database and WIJ. This also enables

direct IO for database read operations.

NOTE: Because direct IO bypasses filesystem cache, OS file copy operations including Caché Online Backup will be VERY slow when direct IO is configured.

For added performance and lowest latency for the WIJ on **RHEL** (this is not supported on SUSE since SUSE 9), and to lessen the impact on other IO we will also configure:

- Use of the rtkaio library for RHEL systems using Caché. **Note: IRIS does not need this library.**

  NOTE: For Caché, Ensemble, and HealthShare distributions beginning with version 2017.1.0. on Linux (only if a backup or async mirror member is configured to use the rtkaio library) you must apply RJF264, available via Ad Hoc distribution from InterSystems Worldwide Response Center (WRC).

Steps

The procedure is to:

1. Shutdown Caché
2. edit the <installdirectory>/cache.cpf file
3. Restart Caché.

In the cache.cpf file add the following three lines to the top of [config] stanza, leaving other lines unchanged, as shown in the example below;

```
[config]
wduseasyncio=1
asyncwij=8
```

For RHEL Caché (not IRIS) also add the following to the [config] section:

```
LibPath=/lib64/rtkaio/
```

Note: When Caché restarts the lines will be sorted in alphabetical order in the [config] stanza.

Summary

This post gave an example of creating a 900GB LVM PE stripe and creating a file system for a database disk on vSAN. To get the best performance from the LVM stripe you also learned how to configure Caché/InterSystems IRIS for asynchronous IO for database writes and the WIJ.

#Best Practices #Deployment #InterSystems Business Solutions and Architectures #Platforms #Red Hat Enterprise Linux (RHEL) #System Administration #Caché #InterSystems IRIS