文章
Michael Lei · 六月 15, 2021 阅读大约需 8 分钟

InterSystems 最佳实践之--LVM PE 条带化使超融合存储吞吐量最大化

本帖概述了通过为 InterSystems 数据平台（InterSystems IRIS、Caché 和 Ensemble）上的数据库磁盘创建 LVM 物理盘区 (PE) 条带来实现低延迟存储 IO 的最佳实践配置，并提供了有用链接。

一致的低延迟存储是获得最佳数据库应用程序性能的关键。 例如，对于在 Linux 上运行的应用程序，经常在数据库磁盘中使用逻辑卷管理器 (LVM)，因为它能够扩展卷和文件系统，或者为在线备份创建快照。 对于数据库应用程序，在使用 LVM PE 条带化逻辑卷的情况下，并行写入还可提高数据 I/O 的效率，从而有助于提高大规模连续读取和写入的性能。

本帖重点介绍在 HCI 中使用 LVM PE 条带，也受到了社区中发布的软件定义的数据中心 (SDDC) 和超融合基础架构 (HCI) – InterSystems 客户的重要注意事项白皮书的启发。 该白皮书推荐 “ 对 Linux 虚拟机使用 LVM PE 条带化，从而将 IO 分布在多个磁盘组 ” 以及 “ 对于 Linux 虚拟机上的所有数据库和写入映像日志 (WIJ) 文件使用异步 IO 及 rtkaio 库 ” 。 本帖提供了这些要求和示例的一些上下文信息。

注：

> 目前有多个超融合、融合和软件定义的供应商平台，我在本帖中不会提供每个平台的详细说明，而是以在 VMware ESXi 和 vSAN 上运行的 Red Hat Enterprise Linux (RHEL) 7.4 上的 InterSystems IRIS 或 Caché 的配置作为示例进行说明。 不过，其他解决方案的基本过程是相似的，特别是在 InterSystems IRIS 或 Caché 和操作系统层面。
> 如果您不确定如何将这些说明转换到其他平台，请联系各供应商的支持人员，了解他们的最佳实践。 InterSystems 技术专家还可以直接向客户和供应商或通过社区提供建议。

还需要注意的是，本帖中关于 LVM PE 条带化的指南既适用于 HCI，也适用于 “ 传统 ” 存储。

是否必须使用 LVM 条带化？

对于磁盘阵列等传统存储，简短的答案是 “ 否 ” 。 对数据库磁盘运行 LVM 条带化卷并不是必需的，尤其是使用现代全闪存阵列的情况下；如果性能尚可，并且您没有 LVM 需求，则无需改动。

但是，如上文所述，建议在 Nutanix 和 VMware VSAN 等超聚合和存储解决方案上的数据库磁盘中使用 LVM 条带，以便在 IO 操作中可以使用更多主机节点和磁盘组。

为什么对数据平台使用 LVM 条带？

特别建议 HCI 上的数据库磁盘使用 LVM 条带，以降低某些架构功能的性能开销，例如减轻写入守护进程 (WD) 对数据库写入和日志写入的影响。 使用 LVM 条带将数据库突发写入分散到更多磁盘设备和多个磁盘组。 此外，本帖还将说明如何增加大规模 IO 写入映像日志 (WIJ) 的并行性，从而减少对其他 IO 的延迟影响。

> 注意：在本帖中，当我提到 “ 磁盘 ” 时，我指的是 NVMe、Optane、SATA 或 SAS SSD，或者任何其他闪存设备。

vSAN 存储架构概述

HCI 存储（例如在 vSAN 上运行 ESXi 时）使用两个磁盘层：一个缓存层和一个容量层。
对于全闪存架构**（必须使用全闪存，不要使用旋转磁盘！）**
，所有写入操作都在缓存层进行，随后数据最终会转移到容量层。
读取来自容量层（也可能来自缓存层上的缓存）。 HCI 集群中的每个主机都可以有一个或多个磁盘组。
在使用磁盘组的情况下（例如使用 VSAN 时），每个磁盘组都由一个缓存磁盘和多个容量磁盘组成。
例如，缓存磁盘是单个 NVMe 磁盘，容量磁盘是三个或更多写密集型 SAS SSD 磁盘。

有关 HCI（包括 vSAN 磁盘组）的更多详细信息，请参见社区上的帖子 超融合基础架构 (HCI) ” 或联系您的 HCI 供应商。

LVM 条带化逻辑卷概述

Red Hat 支持网站上提供了很好的 Linux LVM 概述，其他地方，例如这里的面向系统管理员的教程也非常好。

数据平台存储 IO

了解 InterSystems 数据平台生成的 IO 类型很重要。 社区中提供了存储 IO 模式的概述。

创建 LVM PE 条带的过程

先决条件和步骤

在我们深入讨论该过程之前，您还应该记住，其他变数也可能影响存储性能。 仅创建 LVM
条带并不能保证实现最佳性能，还必须考虑存储类型，以及整个 IO 路径，包括 IO 队列和队列深度。

本示例适用于 VMware，您还应该阅读 InterSystems IRIS VMware 最佳实践指南，并应用其中的建议。
尤其是存储方面的注意事项，例如跨 PVSCSI 控制器分离存储 IO 类型。

概述

以下示例展示了在 VMware ESXi 和 VSAN 6.7 上运行的 Red Hat Enterprise Linux (RHEL) 7.4 上使用 InterSystems IRIS 或 Caché 的最佳实践。

下文介绍以下步骤：

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

---

1. ESXi 配置

a) 创建 VMDK 磁盘

必须按照 InterSystems IRIS VMware 最佳实践指南创建磁盘；数据库、日志和 WIJ 在不同的 PVSCI 设备上。

创建的 VMDK 数量取决于您的规模调整要求。 在本示例中，数据库文件系统将由四个 255 GB VMDK
磁盘组成，这些磁盘将一起条带化，为数据库文件系统创建一个 900GB 逻辑磁盘。

步骤：

1. 在添加 VMDK 前关闭虚拟机。
2. 在 vCenter 控制台中创建多个磁盘 (VMDK)，每个磁盘为 255GB，单个 LVM 条带中的所有磁盘都必须与同一个 PVSCSI 控制器关联。
3. 启动虚拟机。 在启动过程中，将在操作系统中创建新磁盘，例如 /dev/sdi 等。

为什么创建多个 255 GB VMDK？在 vSAN 中，存储组件以 256 GB 区块为单位创建，我们将 VMDK 大小保持在恰好低于 256 GB，是为了强制使组件位于不同的磁盘组上。
从而实施另一个层面的条带化（在我的测试中是这样，但我不保证 vSAN 实际也是如此）。

注意：在创建过程中，vSAN 将磁盘组件分布到所有主机和磁盘组，以确保可用性。 例如，在允许的故障数 (FTT) 设置为 2 的情况下，每个磁盘组件有三个副本，加上两个小的见证组件，全部都在不同的主机上。
如果磁盘组、主机或网络发生故障，应用程序将使用其余磁盘组件继续运行，而不会丢失数据。
我们对这个过程可能多虑了！ 在 vSAN 等 HCI 解决方案中，无法控制组成 VMDK 的组件在某个时间点位于哪个物理磁盘上。 事实上，由于维护、重新同步或重建的原因，随着时间的推移，VMDK 可能会移动到不同的磁盘组或主机上。 这是正常的。

---

2. RHEL 配置

a) 确认对于每个磁盘设备，RHEL IO 调度器都为 NOOP。

最佳实践是使用 ESXi 内核的调度器。 有关设置调度器的更多信息，请参见 Red Hat 知识库文章。
我们建议在启动时为所有设备设置该选项。 要验证是否已正确设置调度器，可以显示磁盘设备（例如，在本例中为 /dev/sdi）的当前设置，如下所示：

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

您可以看到 noop 已启用，因为它放在方括号中突出显示。

b) 创建条带化的 LVM 和 XFS 文件系统

现在，我们准备在 RHEL 中创建 LVM 条带和数据库文件系统。
以下是所涉及步骤的示例，请注意，对于您的环境，需要替换虚构的名称 vgmydb、lvmydb01 和路径 /mydb/db。

步骤

1. 使用 vgcreate 命令创建带有新磁盘设备的卷组。

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

例如，如果创建磁盘 /dev/sdh、/dev/sdi、/dev/sdj 和 /dev/sdk：

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

2. 使用 lvcreate 命令创建条带化逻辑卷。 建议至少四个磁盘。 从 4MB 条带开始，但是对于非常大的逻辑卷，系统可能会提示您选择更大的大小，如 16M。

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

例如，要创建带有 4 个条带的 900GB 磁盘，且条带大小为 4 MB：

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

3. 使用 mkfs 命令创建数据库文件系统。

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

例如：

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

4. 创建文件系统挂载点，例如：

```
mkdir /mydb/db
```

5. 编辑 /etc/fstab 以添加以下挂载条目并挂载文件系统。 例如：

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

6. 挂载新的文件系统。

```
mount /mydb/db
```

3. Caché/InterSystems IRIS 配置

本节我们将配置：

- 异步和直接 IO，以实现数据库和 WIJ 的最佳写入性能。 这也为数据库读取操作启用了直接 IO。

  注意：由于直接 IO 会绕过文件系统缓存，因此在配置直接 IO 后，操作系统文件复制操作（包括 Caché 在线备份）将非常慢。

为提高 **RHEL** 上的 WIJ 的性能并实现最低延迟（SuSE 9 及更高版本不支持），并减少对其他 IO 的影响，我们还将配置：

- 将 rtkaio 库用于使用 Caché 的 RHEL 系统。 **注意：IRIS 不需要这个库。**

  注：对于 Linux 上版本号以 2017.1.0. 开头的 Caché、Ensemble 和 HealthShare 发行版（仅当备份或异步镜像成员配置为使用 rtkaio 时），必须应用 RJF264，可通过 InterSystems 全球响应中心 (WRC) 的特别分发获取。

步骤

步骤为：

1. 关闭 Caché
2. 编辑 <installdirectory>/cache.cpf 文件
3. 重启 Caché

在 cache.cpf 文件中，将以下三行添加到 [config] 部分的顶部，其他行保持不变，如下例所示：

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

对于 RHEL Caché（不是 IRIS），还要将以下内容添加到 [config] 部分：

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

注意：当 Caché 重新启动后，这些行将在 [config] 部分中按字母顺序排序。

总结

本帖给出了创建 900GB LVM PE 条带和为 vSAN 上的数据库磁盘创建文件系统的示例。 为了通过 LVM 条带获得最佳性能，您还学习了如何配置 Caché/InterSystems IRI 来为数据库写入和 WiJ 启用异步 IO。

#InterSystems 业务解决方案和架构 #Red Hat Enterprise Linux (RHEL) #平台 #系统管理 #部署 #Caché #InterSystems IRIS

源 URL: https://cn.community.intersystems.com/post/intersystems-%E6%9C%80%E4%BD%B3%E5%AE%9E%E8%B7%B5%E4%B9%8B-lvm-pe-%E6%9D%A1%E5%B8%A6%E5%8C%96%E4%BD%BF%E8%B6%85%E8%9E%8D%E5%90%88%E5%AD%98%E5%82%A8%E5%90%9E%E5%90%90%E9%87%8F%E6%9C%80%E5%A4%A7%E5%8C%96