40 Redis的下一步：基于NVM内存的实践

今天这节课是咱们课程的最后一节课了，我们来聊聊 Redis 的下一步发展。

这几年呢，新型非易失存储（Non-Volatile Memory，NVM）器件发展得非常快。NVM 器件具有容量大、性能快、能持久化保存数据的特性，这些刚好就是 Redis 追求的目标。同时，NVM 器件像 DRAM 一样，可以让软件以字节粒度进行寻址访问，所以，在实际应用中，NVM 可以作为内存来使用，我们称为 NVM 内存。

你肯定会想到，Redis 作为内存键值数据库，如果能和 NVM 内存结合起来使用，就可以充分享受到这些特性。我认为，Redis 发展的下一步，就可以基于 NVM 内存来实现大容量实例，或者是实现快速持久化数据和恢复。这节课，我就带你了解下这个新趋势。

接下来，我们先来学习下 NVM 内存的特性，以及软件使用 NVM 内存的两种模式。在不同的使用模式下，软件能用到的 NVM 特性是不一样的，所以，掌握这部分知识，可以帮助我们更好地根据业务需求选择适合的模式。

NVM 内存的特性与使用模式

Redis 是基于 DRAM 内存的键值数据库，而跟传统的 DRAM 内存相比，NVM 有三个显著的特点。

首先，**NVM 内存最大的优势是可以直接持久化保存数据。**也就是说，数据保存在 NVM 内存上后，即使发生了宕机或是掉电，数据仍然存在 NVM 内存上。但如果数据是保存在 DRAM 上，那么，掉电后数据就会丢失。

其次，**NVM 内存的访问速度接近 DRAM 的速度。**我实际测试过 NVM 内存的访问速度，结果显示，它的读延迟大约是 200~300ns，而写延迟大约是 100ns。在读写带宽方面，单根 NVM 内存条的写带宽大约是 1~2GB/s，而读带宽约是 5~6GB/s。当软件系统把数据保存在 NVM 内存上时，系统仍然可以快速地存取数据。

最后，**NVM 内存的容量很大。**这是因为，NVM 器件的密度大，单个 NVM 的存储单元可以保存更多数据。例如，单根 NVM 内存条就能达到 128GB 的容量，最大可以达到 512GB，而单根 DRAM 内存条通常是 16GB 或 32GB。所以，我们可以很轻松地用 NVM 内存构建

TB 级别的内存。

总结来说，NVM 内存的特点可以用三句话概括：

- 能持久化保存数据；
- 读写速度和 DRAM 接近；
- 容量大。

现在，业界已经有了实际的 NVM 内存产品，就是 Intel 在 2019 年 4 月份时推出的 Optane AEP 内存条（简称 AEP 内存）。我们在应用 AEP 内存时，需要注意的是，AEP 内存给软件提供了两种使用模式，分别对应着使用了 NVM 的容量大和持久化保存数据两个特性，我们来学习下这两种模式。

第一种是 Memory 模式。

这种模式是把 NVM 内存作为大容量内存来使用的，也就是说，只使用 NVM 容量大和性能高的特性，没有启用数据持久化的功能。

例如，我们可以在一台服务器上安装 6 根 NVM 内存条，每根 512GB，这样我们就可以在单台服务器上获得 3TB 的内存容量了。

在 Memory 模式下，服务器上仍然需要配置 DRAM 内存，但是，DRAM 内存是被 CPU 用作 AEP 内存的缓存，DRAM 的空间对应用软件不可见。换句话说，**软件系统能使用到的内存空间，就是 AEP 内存条的空间容量。**

第二种是 App Direct 模式。

这种模式启用了 NVM 持久化数据的功能。在这种模式下，应用软件把数据写到 AEP 内存上时，数据就直接持久化保存下来了。所以，使用了 App Direct 模式的 AEP 内存，也叫做持久化内存（Persistent Memory，PM）。

现在呢，我们知道了 AEP 内存的两种使用模式，那 Redis 是怎么用的呢？我来给你具体解释一下。

基于 NVM 内存的 Redis 实践

当 AEP 内存使用 Memory 模式时，应用软件就可以利用它的大容量特性来保存大量数据，Redis 也就可以给上层业务应用提供大容量的实例了。而且，在 Memory 模式下，Redis 可以像在 DRAM 内存上运行一样，直接在 AEP 内存上运行，不用修改代码。

不过，有个地方需要注意下：在 Memory 模式下，AEP 内存的访问延迟会比 DRAM 高一点。我刚刚提到过，NVM 的读延迟大约是 200~300ns，而写延迟大约是 100ns。所以，在 Memory 模式下运行 Redis 实例，实例读性能会有所降低，我们就需要在保存大量数据和读性能较慢两者之间做个取舍。

那么，当我们使用 App Direct 模式，把 AEP 内存用作 PM 时，Redis 又该如何利用 PM 快速持久化数据的特性呢？这就和 Redis 的数据可靠性保证需求和现有机制有关了，我们来具体分析下。

为了保证数据可靠性，Redis 设计了 RDB 和 AOF 两种机制，把数据持久化保存到硬盘上。

但是，无论是 RDB 还是 AOF，都需要把数据或命令操作以文件的形式写到硬盘上。对于 RDB 来说，虽然 Redis 实例可以通过子进程生成 RDB 文件，但是，实例主线程 fork 子进程时，仍然会阻塞主线程。而且，RDB 文件的生成需要经过文件系统，文件本身会有一定的操作开销。

对于 AOF 日志来说，虽然 Redis 提供了 always、everysec 和 no 三个选项，其中，always 选项以 fsync 的方式落盘保存数据，虽然保证了数据的可靠性，但是面临性能损失的风险。everysec 选项避免了每个操作都要实时落盘，改为后台每秒定期落盘。在这种情况下，Redis 的写性能得到了改善，但是，应用会面临秒级数据丢失的风险。

此外，当我们使用 RDB 文件或 AOF 文件对 Redis 进行恢复时，需要把 RDB 文件加载到内存中，或者是回放 AOF 中的日志操作。这个恢复过程的效率受到 RDB 文件大小和 AOF 文件中的日志操作多少的影响。

所以，在前面的课程里，我也经常提醒你，不要让单个 Redis 实例过大，否则会导致 RDB 文件过大。在主从集群应用中，过大的 RDB 文件就会导致低效的主从同步。

我们先简单小结下现在 Redis 在涉及持久化操作时的问题：

- RDB 文件创建时的 fork 操作会阻塞主线程；
- AOF 文件记录日志时，需要在数据可靠性和写性能之间取得平衡；
- 使用 RDB 或 AOF 恢复数据时，恢复效率受 RDB 和 AOF 大小的限制。

但是，如果我们使用持久化内存，就可以充分利用 PM 快速持久化的特点，来避免 RDB 和 AOF 的操作。因为 PM 支持内存访问，而 Redis 的操作都是内存操作，那么，我们就可以把 Redis 直接运行在 PM 上。同时，数据本身就可以在 PM 上持久化保存了，我们就不再需要额外的 RDB 或 AOF 日志机制来保证数据可靠性了。

那么，当使用 PM 来支持 Redis 的持久化操作时，我们具体该如何实现呢？

我先介绍下 PM 的使用方法。

当服务器中部署了 PM 后，我们可以在操作系统的 /dev 目录下看到一个 PM 设备，如下所示：

```
/dev/pmem0
```

然后，我们需要使用 ext4-dax 文件系统来格式化这个设备：

```
mkfs.ext4 /dev/pmem0
```

接着，我们把这个格式化好的设备，挂载到服务器上的一个目录下：

```
mount -o dax /dev/pmem0  /mnt/pmem0
```

此时，我们就可以在这个目录下创建文件了。创建好了以后，再把这些文件通过内存映射（mmap）的方式映射到 Redis 的进程空间。这样一来，我们就可以把 Redis 接收到的数据直接保存到映射的内存空间上了，而这块内存空间是由 PM 提供的。所以，数据写入这块空间时，就可以直接被持久化保存了。

而且，如果要修改或删除数据，PM 本身也支持以字节粒度进行数据访问，所以，Redis 可以直接在 PM 上修改或删除数据。

如果发生了实例故障，Redis 宕机了，因为数据本身已经持久化保存在 PM 上了，所以我们可以直接使用 PM 上的数据进行实例恢复，而不用再像现在的 Redis 那样，通过加载 RDB 文件或是重放 AOF 日志操作来恢复了，可以实现快速的故障恢复。

当然，因为 PM 的读写速度比 DRAM 慢，所以，**如果使用 PM 来运行 Redis，需要评估下 PM 提供的访问延迟和访问带宽，是否能满足业务层的需求。**

我给你举个例子，带你看下如何评估 PM 带宽对 Redis 业务的支撑。

假设业务层需要支持 1 百万 QPS，平均每个请求的大小是 2KB，那么，就需要机器能支持 2GB/s 的带宽（1 百万请求操作每秒 * 2KB 每请求 = 2GB/s）。如果这些请求正好是写操作的话，那么，单根 PM 的写带宽可能不太够用了。

这个时候，我们就可以在一台服务器上使用多根 PM 内存条，来支撑高带宽的需求。当然，我们也可以使用切片集群，把数据分散保存到多个实例，分担访问压力。

好了，到这里，我们就掌握了用 PM 将 Redis 数据直接持久化保存在内存上的方法。现在，我们既可以在单个实例上使用大容量的 PM 保存更多的业务数据了，同时，也可以在实例故障后，直接使用 PM 上保存的数据进行故障恢复。

小结

这节课我向你介绍了 NVM 的三大特点：性能高、容量大、数据可以持久化保存。软件系统可以像访问传统 DRAM 内存一样，访问 NVM 内存。目前，Intel 已经推出了 NVM 内存产品 Optane AEP。

这款 NVM 内存产品给软件提供了两种使用模式，分别是 Memory 模式和 App Direct 模式。在 Memory 模式时，Redis 可以利用 NVM 容量大的特点，实现大容量实例，保存更多数据。在使用 App Direct 模式时，Redis 可以直接在持久化内存上进行数据读写，在这种情况下，Redis 不用再使用 RDB 或 AOF 文件了，数据在机器掉电后也不会丢失。而且，实例可以直接使用持久化内存上的数据进行恢复，恢复速度特别快。

NVM 内存是近年来存储设备领域中一个非常大的变化，它既能持久化保存数据，还能像内存一样快速访问，这必然会给当前基于 DRAM 和硬盘的系统软件优化带来新的机遇。现在，很多互联网大厂已经开始使用 NVM 内存了，希望你能够关注这个重要趋势，为未来的发展做好准备。

每课一问

按照惯例，我给你提个小问题，你觉得有了持久化内存后，还需要 Redis 主从集群吗?

欢迎在留言区写下你的思考和答案，我们一起交流讨论。如果你觉得今天的内容对你有所帮助，也欢迎你分享给你的朋友或同事。

上一页 下一页