

简易 DMA 解压样本

*\*此样本与 2017 年 6 月的 Xbox One XDK 兼容*

# 描述



# 此样本演示了如何在几种不同的方案中使用硬件 LZ 压缩和解压功能。

# Xbox One GPU 允许直接访问实现 DEFLATE 算法的 DMA 引擎 (<http://www.rfc-base.org/rfc-1951.html>) 以及 Jpeg Decode。 这些功能可用于卸载（如果不卸）会占用大量 CPU 的任务。

# 使用样本

## 此样本将使用以下控件：

|  |  |
| --- | --- |
| 操作 | 游戏手柄 |
| 通过载入纹理循环 | 方向键上/下 |
| 切换硬件/软件压缩 | A 键 |
| 切换硬件/软件解压 | B 键 |
| 退出 | 查看按键 |

# 实施说明

## 场景 1：运行时解码/编码

加载样本时，将拾取样本的媒体\纹理\文件夹中存在的任何 DDS 纹理文件。 可通过修改 MediaList.txt 文件并重新部署来添加其他文件。

加载后，可使用 Dma 硬件（带 1KB 窗口的 Lz77）或使用参考压缩库 Zlib 或 Zopfli 在压缩之间进行更改。 使用后者软件压缩选项时，压缩仅限于解压引擎支持的使用 4KB 窗口的最优选项。 在这些选项之间切换可轻松比较可用的压缩选项。 另请注意，较大的文件会碎片化为略低于 4MB 的区块，以保持与硬件的兼容性。

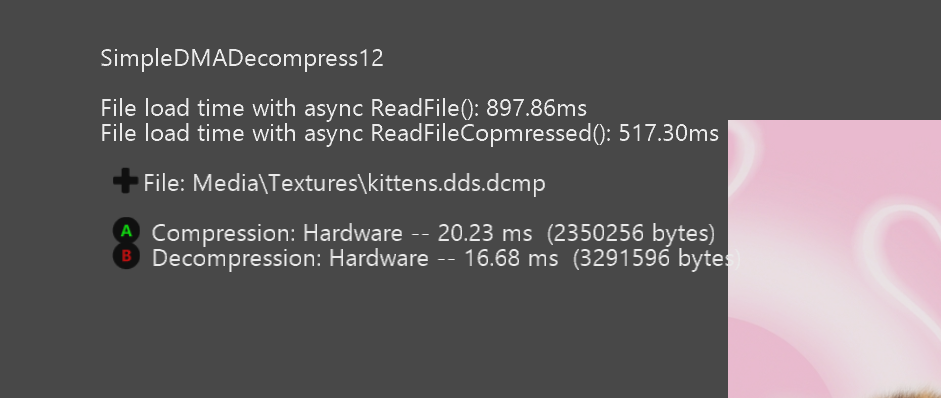
## 场景 2：流媒体 DMA 解压

这种场景就像大多数主题所做的那样，解压从磁盘流入、作为内容构建管道的一部分而进行压缩的内容。 在此变体中，所有解压工作都卸载到专用硬件，而非消耗 CPU 周期。

为了实现此目的，StreamingDmaCompressionLib 项目将定义一个由压缩区块组成的简易文件格式，让异步能够卸载到 DMA 硬件，以提高有效 I/O 吞吐量并减小标题的打包尺寸。 为了便于使用，XboxDmaCompression 工具（仅限 PC）允许以硬件支持的最佳方式压缩整个文件。 或者，可直接使用库，以便以流级进行压缩，能够将压缩内容嵌入到更大的复合文件中。

使用 ReadFileCompressed() 读取生成的文件（或嵌入的压缩流），与 Win32 ReadFile() api 基本上为 API 兼容。 适用于重叠的异步 I/O，支持缓冲和非缓冲读取，后者可提高性能。

为了演示此系统的吞吐量特性，默认的 MediaList.txt 复制包含纹理的 20 个压缩副本和 20 个未压缩副本，在初始化\加载路径中一次加载一个批处理。 每批未压缩的总字节数为 59.6MB，未压缩负载 (66Mb/s) 和压缩负载 (115Mb/s) 的有效吞吐量大不相同。 结合 DmaCompressionTool，这是一种用于确定 DMA 解压硬件是否对标题有意义的简易方法。 库为完全异步，因此与底层 Windows 重叠\同步 I/O 规模类似，随着请求数量的增加而改进。



使用 ReadFileCompressed() api 时，主要区别在于在调用 API 时不知道解压缓冲区的尺寸。 并非要求标题事先了解这些信息，因为这需要一些其他的阻塞读取，API 支持传递自定义分配器，或者可使用默认分配器，加载包含文件标题的区块后将完成分配。

包括 ReadFileCompressed 的重叠 I/O 的其他方面与 ReadFile 相同，但有一些例外。 ReadFileCompressed 可以处理未缓冲文件上的非扇区对齐读取，在内部处理预\后缓冲。 但是，ReadFileCompressed 要求读取 4 字节对齐，以确保 dma 从存储控制器移动，且 dma 通过解压器移动可在 CPU 未复制任何内存的情况下完成。 如果使用 StreamingDmaCompressionLib 压缩较大的复合文件中的流媒体，必须确保压缩流媒体的第一个字节在文件中以 4 字节对齐。

# 已知的问题

* 此样本未正确实现暂停/恢复。

# 更新历史记录

此样本的原始版本编写于 2012 年。于 2018 年 9 月进行重写。

# 隐私声明

编译和运行示例时，示例可执行文件的文件名将发送给Microsoft以帮助跟踪示例使用情况。要选择退出此数据收集，您可以删除Main.cpp中标记为“Sample Usage Telemetry”的代码块。

有关 Microsoft 隐私政策的更多信息，请参阅 [Microsoft 隐私声明](https://privacy.microsoft.com/zh-cn/privacystatement/)。