## 优化实验结果与分析

### 哈达玛失真函数并行优化实验结果与分析

首先分析帧内预测模块种对于哈达玛失真函数的并行优化结果。使用与前文测试中相同的测试环境和视频测试序列。其中，由于前文提出了2种并行优化方法，因此实验中对两种实现都进行了测试。结果如下图所示：

图中HADs函数表示HM中实现的串行哈达玛失真函数，CUDA\_HADs\_Z表示用“依赖整合法”实现的并行函数，CUDA\_HADs\_D表示用“代入展开法”实现的并行函数。通过实验可以发现，两种方法相对于串行函数都有一定的加速效果，且对于不同测试序列优化性能基本稳定。结合对于理论并行度的计算，这两个函数的加速性能如下所示：

|  |  |  |
| --- | --- | --- |
| 函数名 | 最大并行度 | 平均加速比 |
| CUDA\_HADs\_Z | 32 | 1.158 |
| CUDA\_HADs\_D | 64 | 1.2655 |

可见使用“带入展开法”实现的并行函数有更好的平均加速比，相比于穿行算法能速度能提升26%左右。而这一结果也和并行度分析一致，因此在之后的帧内预测模块整体优化和HM整体测试中，选择使用该方法来进行哈达玛失真的并行计算。

### 预测值函数并行优化实验结果与分析

接下来分析帧内预测模块中对于预测值计算函数的并行优化结果。结果如下图所示：

图中predIntraAng表示HM中的串行预测值计算函数，CUDA\_PIA表示并行优化后的预测值计算函数。如图所示，该部分优化对于不同测试序列也保持性能稳定，其平均加速比约为11%。

### 帧内预测模块整体并行优化实验结果与分析

对于帧内预测模块的整体优化包含了对于以上2类函数的多线程并行调用，因此在结果统计的过程中，一方面记录了并行优化前后模块的整体的运行时间，另一方面也记录了哈达玛失真函数和预测值计算函数在高并发调用的情况下的优化情况，以便更好地分析优化后的帧内预测模块中具体函数的性能提升。模块整体的运行时间与其中优化函数的运行时间实验数据如下：

根据这一结果，可以发现，经过帧内模块的整体优化之后，哈达玛失真函数和预测值计算函数的加速性能相比于之前有巨大提升。这一性能的提升主要收益于更高的并行度。如前文所述，在帧内预测模块的整体优化中，哈达玛失真函数的并行度最大提高64×35倍，预测值计算函数的并行度最大提高35倍。在模块并行的条件下，这两个函数的平均加速比和理论最大并行度如下所示：

|  |  |  |
| --- | --- | --- |
| 函数名 | 理论并行度 | 平均加速比 |
| CUDA\_HADs | 64 | 1.2655 |
| CUDA\_HADs\_MA | 143360 | 9.1975 |
| CUDA\_PIA | 4096 | 1.1095 |
| CUDA\_PIA\_MA | 143360 | 1.7836 |
| CUDA\_INTRA\_MA | 143360 | 1.2207 |

表中\_MA后缀表示“模块加速”（module acceleration），即在帧内预测模块整体优化情况下的对应函数，最后一行的CUDA\_INTRA\_MA表示整个帧内预测模块。其中哈达玛失真函数的加速比达到了9.2左右，而预测值计算函数的加速比达到了1.8，相比于之前单一函数的性能提升更为显著。因此整个帧内预测模块的加速比也达到了1.22以上。根据前文对函数运行时间和所占比重的分析，哈达玛失真函数占帧内预测模块运行时间的约13%，预测值计算函数占约12%，可知对模块优化的最大理论加速比为1/（1-25%）≈1.33。对比实际实现的并行方法，已经达到了理论最大加速比的91%，基本符合预期。

### 快速DST函数并行优化实验结果与分析

下面对变换量化模块的优化结果进行实验和分析。首先分析快速DST变换的优化结果，实验结果如下所示：

图中FDST表示HM中实现的串行快速DST变换函数，CUDA\_FDST表示并行优化之后的快速DST函数。可以发现，该算法的优化性能对于不同的测试序列也保持了基本稳定。根据前文分析，该算法的最大理论并行度为16，实际算法的加速比为1.36。

### 蝶形DCT变换函数并行优化实验结果与分析

对于蝶形DCT变换，也进行了两种方法的并行优化，即“单一核函数法”和“多阶段核函数法”，首先对这两种方法进行实验和分析，选择其中较好的的并行实现方法作为最终的优化函数。以32×32大小的蝶形DCT变换函数为例，这两种方法的运行时间分别为：

其中PBF32（Partial Butterfly 32）为HM中的串行蝶形变换函数，CUDA\_PBF32\_SK为单一核函数法优化的并行函数，CUDA\_PBF32\_MS为多阶段核函数法优化的并行函数。从图中可以看出，这两种方法对于原函数都有较好的加速效果，且加速效果对于不同的测试序列基本保证稳定。2种方法的具体加速比和理论最大并行度如下：

|  |  |  |
| --- | --- | --- |
| 函数名 | 最大并行度 | 平均加速比 |
| CUDA\_PBF32\_SK | 512 | 8.1153 |
| CUDA\_PBF32\_MS | 1024 | 5.505 |

从表中可以发现单核函数法的实际优化效果更好，而其理论最大并行度却比多阶段核函数法要低。这是主要是因为在多阶段和函数法种，调用了多次核函数，虽然最大线程数更多，但是单线程的计算方法过于简单，程序在线程同步和设备通信上的耗时反而较高。而单一核函数法只需要调用一个核函数，且每一个线程具有一定的计算深度，从而充分利用了线程计算能力。经过这一实验，选择单一核函数方法来作为最终的蝶形DCT变换并行优化方法。

选择较佳的方法之后，对于不同矩阵大小的蝶形DCT变换函数分别实现其并行函数并进行试验，具体结果如下：

从图中可以发现，单一核函数方法对于不同尺寸的蝶形DCT变换函数都有比较明显的优化效果，并且随着输入矩阵的尺寸从4到32，并行函数的加速加速比也逐渐增加。具体每种尺寸的变换函数优化之后的实际加速比和理论最大并行度如下：

|  |  |  |
| --- | --- | --- |
| 函数名 | 理论并行度 | 平均加速比 |
| CUDA\_PBF32\_SK | 512 | 8.1153 |
| CUDA\_PBF16\_SK | 128 | 1.7448 |
| CUDA\_PBF8\_SK | 64 | 1.4106 |
| CUDA\_PBF4\_SK | 8 | 1.2092 |

从中可以发现，对于尺寸最大的32×32矩阵的蝶形DCT变换，其加速比达到了8.1以上，而对于较小的尺寸则加速效果相对较小，这一结果与理论并行度也是一致的。

完成DCT变换相关函数的优化和实验之后，可以对整个变换量化模块的优化结果进行分析。根据前文所述，变换量化函数中的快速DST变换和蝶形DCT变换总计占到了整个模块运行时间的50%以上，因此文中涉及的优化部分，理论最大加速比为1/（1-50%）=2。而实际实现的优化加速比为1.25，达到了最大加速比的62%，基本符合预期。

### HM整体并行优化实验结果与分析

完成对于帧内预测模块和变化量化模块的实验与分析之后，可以看出，并行优化的实现对于这两个模块分别取得了符合预期的效果，因而下面接着测试HM整体在经过并行优化之后的运行情况。在测试时，通过特性宏CUDA\_ACC来控制工程中各处的CUDA函数启用，其中经过前文实验的选择，哈达玛失真函数使用“依赖展开法”实现，蝶形DCT变换使用“单核函数法”实现，且两个模块的优化同时开启。具体运行情况如下：

从图中可以看出，在不同测试序列下，并行优化对于HM整体的加速效果是比较稳定的，平均加速比达到了1.19。根据文中所优化部分，HM整体的理论最大加速比为1.33，实际加速比达到了理论加速比的89%，基本达到了并行优化设计的要求。

### 本章小结