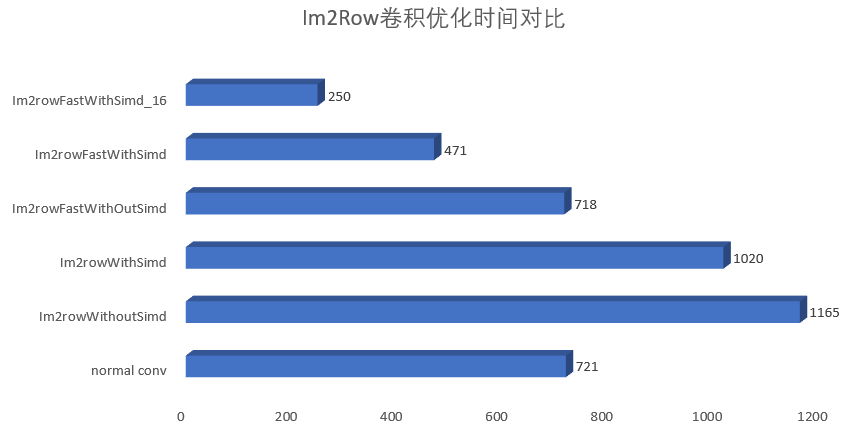
为了方便计算 stride为1，input与kernel都是方形且只有一层,InputSize为500，KernelSize为16



以上是Im2Row卷积优化时间对比图，可以看出对于传统的Im2Row虽然空间上连续提高cache命中率并且使用Simd（可以同时对多个数进行运算）优化，速度还是不如直接卷积，因为转换大量赋值的时间延迟相比以上优点占比更重。对于快速Im2Row，它除了具备空间连续提高cache命中率可以使用Simd外，还减少了大量赋值转换所需要的时间，在不使用Simd时速度和直接卷积差不多，说明转换的时间缺点和空间连续cache命中高的优点几乎抵消，使用Simd后比直接卷积快34%，接着我将快速Im2Row的数据类型变成fp16，速度又比fp32快了近一倍，符合预期。

传统的Im2Row速度不如直接卷积，快速Im2Row在不使用Simd时速度和直接卷积差不多，使用Simd后比直接卷积快34%，将快速Im2Row的数据类型变成fp16，速度又比fp32快了近一倍，符合预期。

为了方便计算 stride为1，input与kernel都是方形且只有一层

I 为InputSize

K为kernelSize

O为outputSize

O=I-（K-1）

传统Im2Row 将Input转为对应矩阵需要复制K²\*O²次

快速Im2Row 将Input转为对应矩阵需要复制（K²+（I-K）\*K）\*O次

计算得出

快速Im2Row复制次数是传统Im2Row的I/K\*(I-K+1)