2023年中国图计算挑战赛

队名：分布式摸鱼

队员：蔡文铠、欧阳建鸿、杨俣哲

比赛简介

主要内容

为图卷积神经网络推理问题的计算优化

推理公式如下

$$X^{(l+1)} = \alpha\left(\hat{A}X^{(l)}W^{(l)}\right)$$

还有激活函数ReLU和LogSoftmax

公式分别为

$$ReLU(x) = max(0, x)$$

$$\mathrm{LogSoftmax}\left(X_{i,j}^{(l)}\right) = \left(X_{i,j}^{(l)} - X_{i,max}^{(l)}\right) - \log\left(\sum_{c=0}^{F_l-1} e^{X_{i,c}^{(l)} - X_{i,max}^{(l)}}\right)$$

$$X_{i,max}^{(l)} = \max\left(X_{i,0}^{(l)}, \ldots, X_{i,F_l-1}^{(l)}\right)$$

任务

需在CPU平台上，对给定数据集，在不损失计算精度（计算的中间过程及其最后结果应全部采用32位浮点数精度）的情况下，以尽可能短的时间完成GCN推理的计算。

数据规模

GCN模型：

$F_0 \leq 128$，$F_1 = 16$，$F_2 \leq 32$

图规模：

顶点\边	<500K	<1M	<5M
<500K	1	1	2
<1M		1	1
<5M			1

主要思路

SIMD优化

由于需要在CPU上进行优化，所以可以考虑向量化指令集操作，利用硬件支持加快计算速度。

根据提供的CPU型号为Intel Xeon Gold 5117 @2.00GHz，是提供AVX512指令集的，所以我们采用AVX512对于部分函数进行SIMD重写实现。

举例对于ReLU的函数如下：

- 原来的代码为

```
void ReLU(int dim, float *X)
{
    for (int i = 0; i < v_num * dim; i++)
        if (X[i] < 0)
            X[i] = 0;
}
```

- 修改后的代码为

```
void ReLU(int dim, float *X)
{
    const int num_elements = v_num * dim;
    int i = 0, align_size = num_elements - (num_elements % 16);
    __m512 zero_vector = _mm512_setzero_ps(), cache_vector, res_vector;
```

```
    for (; i < align_size; i += 16) {
        cache_vector = _mm512_loadu_ps(X + i);
        res_vector = _mm512_max_ps(cache_vector, zero_vector);
        _mm512_storeu_ps(X + i, res_vector);
    }
    if (num_elements % 16) {
        __mmask16 mask = (1 << (num_elements % 16)) - 1;
        cache_vector = _mm512_maskz_loadu_ps(mask, X + i);
        res_vector = _mm512_maskz_max_ps(mask, cache_vector, zero_vector);
        _mm512_mask_storeu_ps(X + i, mask, res_vector);
    }
}
```

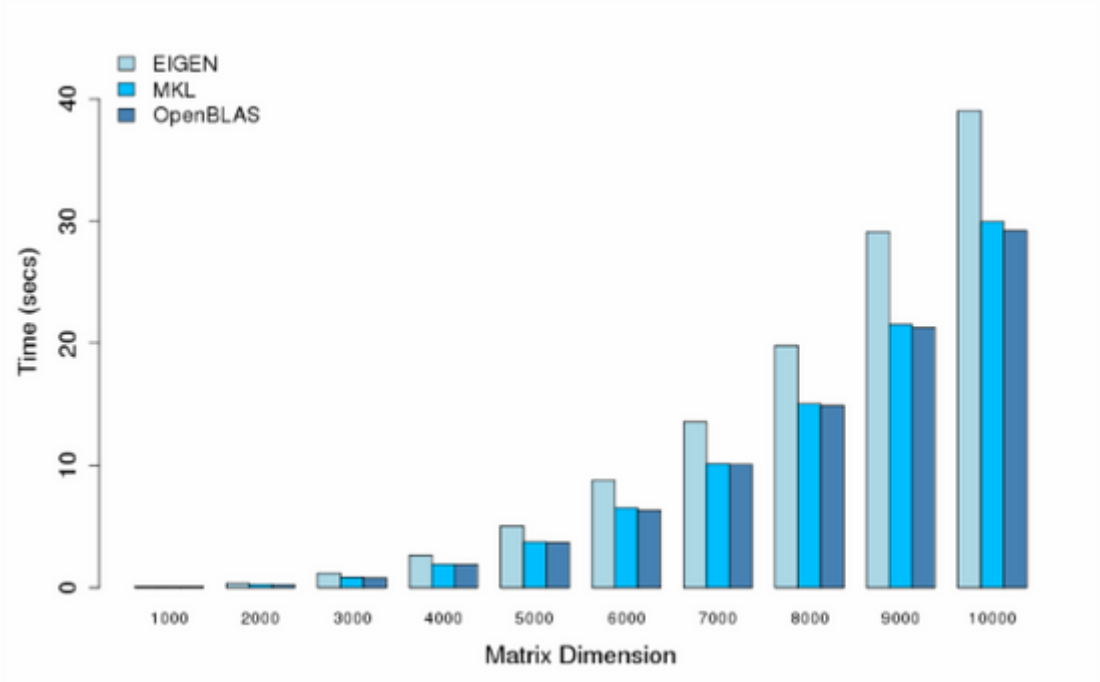
快速矩阵相乘

这里我们使用了开源的OpenBLAS库

我们可以发现对于XW函数，其实就是两个矩阵相乘，而对于矩阵相乘操作目前已经有许多的研究和优化实现，其中比较著名就是OpenBLAS库，我们直接使用它来应用在XW操作上面，可以自适应硬件条件，尽量利用CPU，提升并行度，达到最大的优化速度。

OpenBLAS

- BLAS，基本线性代数子程序，Basic Linear Algebra Subprograms，是一个API标准，用以规范发布 基础线性代数操作的数值库（如矢量或矩阵乘法）
- OpenBLAS是一个开源的矩阵计算库，包含了诸多的精度和形式的矩阵计算算法。就精度而言，包括float和double，两种数据类型的数据，其矩阵调用函数也是不一样。不同矩阵，其计算方式也是有所不同，（姑且认为向量也是一维矩阵），例如，向量与向量之间的计算，向量与矩阵之间的计算，矩阵与矩阵之间的计算。
- Openblas在编译时根据目标硬件进行优化，生成运行效率很高的程序或者库。



在32核下OpenBLAS与其他库的效果比较

应用

- 原代码

```
void XW(int in_dim, int out_dim, float *in_X, float *out_X, float *W)
{
    float(*tmp_in_X)[in_dim] = (float(*)[in_dim])in_X;
    float(*tmp_out_X)[out_dim] = (float(*)[out_dim])out_X;
    float(*tmp_W)[out_dim] = (float(*)[out_dim])W;

    for (int i = 0; i < v_num; i++)
    {
        for (int j = 0; j < out_dim; j++)
        {
            for (int k = 0; k < in_dim; k++)
            {
                tmp_out_X[i][j] += tmp_in_X[i][k] * tmp_W[k][j];
            }
        }
    }
}
```

- 应用OpenBLAS的代码

```
void XW(int in_dim, int out_dim, float *in_X, float *out_X, float *W)
{
```

```
    cblas_sgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans, v_num, out_dim, in_di
          1.0, in_X, in_dim, W, out_dim, 0.0, out_X, out_dim);
}
```

OMP

OpenMP(Open Multi-Processing)是一种共享内存编程模式，多线程并行应用程序界面，使用C，C++语言。由两种形式实现并行功能：编译指导语句和运行时库函数。编译指导语句告诉程序何时开始并行，库函数用来设置线程数及实现其它并行功能。

应用

- 原代码

```
void edgeNormalization()
{
    for (int i = 0; i < v_num; i++)
    {
        for (int j = 0; j < edge_index[i].size(); j++)
        {
            float val = 1 / sqrt(degree[i]) / sqrt(degree[edge_index[i][j]]);
            edge_val[i].push_back(val);
        }
    }
}
```

- 应用OMP

```
void edgeNormalization()
{
    #pragma omp parallel for
    for (int i = 0; i < v_num; i++)
    {
        for (int j = 0; j < edge_index[i].size(); j++)
        {
            float val = 1 / sqrt(degree[i]) / sqrt(degree[edge_index[i][j]]);
            edge_val[i].push_back(val);
        }
    }
}
```

优化效果

Benchmakr 设计

我们使用benchmark的方式，来对推理过程中所有的函数进行时间的计算。

并且使用RMAT库随机生成图数据集进行不同规模的测试比较。

如下图所示，就是在48核上跑的不同规模的时间结果图。

```
Benchmark                                                                Time             CPU   Iterations Edge Norm  Layer1 AX  Layer1 XW  Layer2 AX  Layer2 XW LogSof
tmax    Max Diff  MaxRowSum Preprocess       ReLU
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------
Origin Implemention Small/4096/4096/64/16/32/manual_time                5.62 ms         6.25 ms          109 0.177049 0.0727086  3.02494 0.0969292  1.04697  0.84
9656          0  0.0510669    0.151379    0.152585
Origin Implemention Small/4096/4096/128/16/32/manual_time               10.1 ms         10.8 ms           70 0.176473    0.0737  7.43428 0.0955136  1.05339  0.86
1293          0  0.0522196     0.14912    0.155234
Origin Implemention Small/4096/16384/64/16/32/manual_time               6.71 ms         8.33 ms          105 0.438763 0.189322  3.02307 0.277116  1.04883  1.0
0295          0  0.0515639    0.436842    0.244837
Origin Implemention Small/4096/16384/128/16/32/manual_time              11.2 ms         12.9 ms           63 0.437712 0.190452  7.41839 0.277799  1.0493  1.0
5496          0  0.0516536    0.431042     0.24602
Origin Implemention Small/4096/65536/64/16/32/manual_time               9.69 ms         15.5 ms           74 1.20538 0.591348  3.06968 1.21867  1.30192  0.79
1789          0  0.0512135     1.17654    0.286596
Origin Implemention Small/4096/65536/128/16/32/manual_time              14.3 ms         20.4 ms           50 1.21731 0.592519  7.6673  1.21814  1.25548  0.89
5552          0  0.0512419     1.12735    0.288844
Origin Implemention Small/16384/16384/64/16/32/manual_time              24.5 ms         27.9 ms           29 0.686431 0.307471  12.082  1.45961  5.16778  3.4
3894          0   0.206307    0.563319    0.617898
Origin Implemention Small/16384/16384/128/16/32/manual_time             41.2 ms         43.9 ms           17 0.685507 0.308945  30.7424 0.421274  4.18248  3.4
3149          0   0.208416    0.559502     0.62481
Origin Implemention Small/16384/65536/64/16/32/manual_time              27.0 ms         33.7 ms           26 1.71433 0.839657  12.351  1.26065  4.18416  3.8
2429          0   0.206568     1.67901    0.932783
Origin Implemention Small/16384/65536/128/16/32/manual_time             44.9 ms         51.9 ms           15 1.7172  0.840767  29.9649 1.26837  4.18272  4.0
5079          0   0.205421     1.67669    0.943987
Origin Implemention Small/65536/65536/64/16/32/manual_time              91.9 ms          103 ms            7 2.80044 1.32519  49.0176 1.87353  17.7558  13.
4759          0   0.845735     2.42212     2.36523
Origin Implemention Small/65536/65536/128/16/32/manual_time             161 ms          184 ms            4 2.81728 1.38275  118.75  1.99983  16.7499  13.
5891          0   0.838645     2.64571     2.39931
OpenBlas Implemention Small/4096/4096/64/16/32/manual_time              105 ms         41.5 ms           9 21.6997 16.6304  4.69426 15.2233  4.33869  18.
6256          0     23.544    0.199069   0.0128923
OpenBlas Implemention Small/4096/4096/128/16/32/manual_time             103 ms         43.2 ms           7 19.025  16.4871  5.52748 15.3785  4.27753  20.
0353          0    22.4807    0.191844   0.0124984
OpenBlas Implemention Small/4096/16384/64/16/32/manual_time             115 ms         46.3 ms           7 16.2698 13.9797  7.71721 18.392  6.01419  26.
2478          0    26.1527    0.536554   0.0134861
OpenBlas Implemention Small/4096/16384/128/16/32/manual_time            104 ms         44.8 ms          11 15.2734 13.2602  7.43004 16.5923  6.2995  21
.547          0    21.7774    0.514573   0.959449
OpenBlas Implemention Small/4096/65536/64/16/32/manual_time             107 ms         44.3 ms          11 9.94315 14.3098  8.96144 17.9973  6.27065  25.
1662          0    22.3151     2.01956   0.0142955
OpenBlas Implemention Small/4096/65536/128/16/32/manual_time            108 ms         48.5 ms          12 9.07449 17.0847  7.19843 19.3878  5.01226  24
.059          0    24.9203     1.53951   0.0143455
OpenBlas Implemention Small/16384/16384/64/16/32/manual_time            80.2 ms         42.5 ms           8 11.0573 13.1814  4.0763  12.8036  5.2609  17.
```

```
OpenBlas Implemention Small/16384/16384/128/16/32/manual_time           91.9 ms         44.1 ms          10 9.66602 14.9812  7.46964 13.8505  4.32853  20.
7286          0    20.0971    0.746358   0.0484117
OpenBlas Implemention Small/16384/65536/64/16/32/manual_time            84.7 ms         50.0 ms          12 7.33771  10.111  7.36143 14.9041  6.37866  17.
0544          0    19.4317     2.06457   0.047391
OpenBlas Implemention Small/16384/65536/128/16/32/manual_time           99.1 ms         51.1 ms           8 9.52108 13.7443  7.88504 13.8982  6.12063  23
.753          0    21.7519     2.37276   0.0504584
OpenBlas Implemention Small/65536/65536/64/16/32/manual_time            103 ms         55.9 ms          15 7.36782 13.0413  8.3518  17.5769  5.20257  23
.717          0     23.997     2.83932   0.967943
OpenBlas Implemention Small/65536/65536/128/16/32/manual_time           52.1 ms         57.7 ms          12 4.04686 4.99849  5.50858 6.50244  7.06833  10.
8047          0     9.7561     3.18407   0.277491
Origin Implemention Standard/400000/400000/128/16/32/iterations:5/manual_time 1071 ms         1205 ms           5 19.3486 15.9427  729.234 50.4588  135.026  81.
9932          0    6.28709      18.49     13.8914
Origin Implemention Standard/400000/800000/128/16/32/iterations:5/manual_time 1122 ms         1299 ms           5 30.0976 23.4525  726.049 65.3074  135.029  87.
9175          0    6.29979     30.7271     17.3289
Origin Implemention Standard/400000/4000000/128/16/32/iterations:5/manual_time 1373 ms         1865 ms           5 96.9705  77.075  726.36  120.163  135.083  85.
5123          0    6.40274     100.692     24.9502
Origin Implemention Standard/800000/800000/128/16/32/iterations:5/manual_time 2187 ms         2455 ms           5 40.2902 31.9258  1.49715k 103.461  270.457  163
.439          0    12.9125     39.6013     27.3975
Origin Implemention Standard/800000/4000000/128/16/32/iterations:5/manual_time 2663 ms         3258 ms           5 129.024 101.484  1.56628k 221.554  271.909  181
.002          0    12.7918     135.448     43.6916
Origin Implemention Standard/4000000/4000000/128/16/32/iterations:5/manual_time 11196 ms        12567 ms           5  276.81 347.553  7.41282k 559.719  1.31366k  813
.789          0      64.05     273.878     133.339
OpenBlas Implemention Standard/400000/400000/128/16/32/iterations:5/manual_time 107 ms         225 ms           5 1.85659 12.5549  11.9439 20.1347  5.29371  22.
1899          0    13.3773     17.0732      2.4341
OpenBlas Implemention Standard/400000/800000/128/16/32/iterations:5/manual_time 115 ms         287 ms           5 2.72832 10.8368  10.9381 20.5603  5.71114  18.
7659          0    14.8861     28.4406     1.99369
OpenBlas Implemention Standard/400000/4000000/128/16/32/iterations:5/manual_time 224 ms         657 ms           5 7.28942  20.668  14.0948 34.4648  8.01503  31.
5605          0    18.4204      86.788     2.93979
OpenBlas Implemention Standard/800000/800000/128/16/32/iterations:5/manual_time 209 ms         387 ms           5 3.73305 33.4096  22.6783 58.5419  7.08491  31.
6398          0    6.50725     38.7746     6.32301
OpenBlas Implemention Standard/800000/4000000/128/16/32/iterations:5/manual_time 299 ms         808 ms           5 9.58258 37.8542  18.6651 64.2403  9.45819  27.
5803          0    5.74571      119.51     6.76606
OpenBlas Implemention Standard/4000000/4000000/128/16/32/iterations:5/manual_time 672 ms         1969 ms           5 21.6433 93.6769  68.1896 117.503  28.2608  41.
2587          0    4.07867     272.013     25.1409
```

SIMD和编译O3优化

主要是对于ReLU进行比较，结果如下图显示

```
Benchmark                                                                                                    Time             CPU   Iteratio
ns  Edge Norm  Layer1 AX  Layer1 XW  Layer2 AX  Layer2 XW LogSoftmax   Max Diff  MaxRowSum Preprocess       ReLU
-----------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------
OpenBlas Implemention Standard/500000/500000/128/16/32/iterations:3/manual_time          107 ms          276 ms
 3    3.93719    7.18656    13.3802    17.9739    6.97382    13.1606          0    17.2164     23.904    2.79333
OpenBlas Implemention Standard/500000/1000000/128/16/32/iterations:3/manual_time         142 ms          347 ms
 3    3.36149    4.42249    14.2565    25.8962    6.18627    27.6545          0     21.161    36.6873    2.44628
OpenBlas Implemention Standard/500000/5000000/128/16/32/iterations:3/manual_time         267 ms          834 ms
 3    8.93699    26.7445    13.2909    44.2505    5.95848     38.169          0    12.1636    114.866    2.44777
OpenBlas Implemention Standard/1000000/1000000/128/16/32/iterations:3/manual_time        231 ms          485 ms
 3    4.76139    31.5299    25.8668    60.1104    7.63303     30.625          0    7.84864    50.6344    11.6526
OpenBlas Implemention Standard/1000000/5000000/128/16/32/iterations:3/manual_time        375 ms         1034 ms
 3     12.756    64.6555    21.1829    77.7815    9.02395    23.5253          0    2.45136    158.759    5.14486
OpenBlas Implemention Standard/5000000/5000000/128/16/32/iterations:3/manual_time        830 ms         2521 ms
 3    24.6277    106.792    88.4291    131.429    35.2706    51.9149          0    4.81476    357.024    29.6867
```

SIMD

```
Benchmark                                                                                                    Time             CPU   Iteratio
ns  Edge Norm  Layer1 AX  Layer1 XW  Layer2 AX  Layer2 XW LogSoftmax   Max Diff  MaxRowSum Preprocess       ReLU
-----------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------
OpenBlas Implemention Standard/500000/500000/128/16/32/iterations:3/manual_time          153 ms          268 ms
 3    2.88408    11.0734     14.613    42.9236    5.52437    34.5965          0    12.9392    23.4984    5.34494
OpenBlas Implemention Standard/500000/1000000/128/16/32/iterations:3/manual_time         174 ms          338 ms
 3    3.26879     14.011    16.2807    42.0623    6.79792    32.2008          0    17.0977    36.9322    5.70474
OpenBlas Implemention Standard/500000/5000000/128/16/32/iterations:3/manual_time         264 ms          827 ms
 3    9.65299      18.96    13.8578    50.9708    6.88959    36.1922          0    8.12334    114.575    4.83104
OpenBlas Implemention Standard/1000000/1000000/128/16/32/iterations:3/manual_time        227 ms          492 ms
 3    4.72197    34.2981    23.8735    63.8032    7.83826     29.495          0    5.54726     50.688    7.03732
OpenBlas Implemention Standard/1000000/5000000/128/16/32/iterations:3/manual_time        365 ms         1024 ms
 3    11.5905    49.0198    24.4468    80.9309    7.71658    25.1021          0    1.09662    157.862    7.11888
OpenBlas Implemention Standard/5000000/5000000/128/16/32/iterations:3/manual_time        830 ms         2521 ms
 3    26.6827    104.429    87.9838    131.492    35.2634    52.2944          0    4.85026    352.547    34.0943
```

编译O3优化

OMP和OpenBLAS

主要是对于XW函数作为比较的对象，结果如下图显示

```
-----------------------------------------------------------------------------------------------------------------------------------------
OpenBlas Implemention Standard/500000/500000/128/16/32/iterations:3/manual_time          153 ms          268 ms
 3    2.88408    11.0734     14.613    42.9236    5.52437    34.5965          0    12.9392    23.4984    5.34494
OpenBlas Implemention Standard/500000/1000000/128/16/32/iterations:3/manual_time         174 ms          338 ms
 3    3.26879     14.011    16.2807    42.0623    6.79792    32.2008          0    17.0977    36.9322    5.70474
OpenBlas Implemention Standard/500000/5000000/128/16/32/iterations:3/manual_time         264 ms          827 ms
 3    9.65299      18.96    13.8578    50.9708    6.88959    36.1922          0    8.12334    114.575    4.83104
OpenBlas Implemention Standard/1000000/1000000/128/16/32/iterations:3/manual_time        227 ms          492 ms
 3    4.72197    34.2981    23.8735    63.8032    7.83826     29.495          0    5.54726     50.688    7.03732
OpenBlas Implemention Standard/1000000/5000000/128/16/32/iterations:3/manual_time        365 ms         1024 ms
 3    11.5905    49.0198    24.4468    80.9309    7.71658    25.1021          0    1.09662    157.862    7.11888
OpenBlas Implemention Standard/5000000/5000000/128/16/32/iterations:3/manual_time        830 ms         2521 ms
 3    26.6827    104.429    87.9838    131.492    35.2634    52.2944          0    4.85026    352.547    34.0943
```

OpenBLAS

Benchmark	Time	CPU	Iterations	Edge Norm	Layer1 AX	Layer1 XW	Layer2 AX	Layer2 XW	LogSoftmax	Max Diff	MaxRowSum	Preprocess	ReLU
OpenBlas Implemention Standard/500000/500000/128/16/32/iterations:3/manual_time	93.6 ms	277 ms	3	2.75149	4.63346	27.0992	11.6526	14.2252	5.18514	0	0.675238	23.9662	3.45125
OpenBlas Implemention Standard/500000/1000000/128/16/32/iterations:3/manual_time	106 ms	343 ms	3	3.38382	2.91369	25.8268	13.7774	14.2167	4.29995	0	0.401259	37.5265	3.67332
OpenBlas Implemention Standard/500000/5000000/128/16/32/iterations:3/manual_time	210 ms	845 ms	3	9.96433	9.41672	27.1558	23.1302	14.3374	4.14435	0	0.427541	117.948	3.08803
OpenBlas Implemention Standard/1000000/1000000/128/16/32/iterations:3/manual_time	201 ms	547 ms	3	4.82551	13.9631	60.0204	22.4384	28.857	10.6855	0	1.10722	51.7196	6.92888
OpenBlas Implemention Standard/1000000/5000000/128/16/32/iterations:3/manual_time	335 ms	1085 ms	3	11.7081	22.4569	59.6079	34.1031	28.9079	8.46605	0	1.06046	162.019	6.90902
OpenBlas Implemention Standard/5000000/5000000/128/16/32/iterations:3/manual_time	1083 ms	2846 ms	3	25.164	61.4355	297.195	102.465	143.748	50.6792	0	4.86407	362.686	34.3427

OMP

最终效果

Benchmark	Time	CPU	Iterations	Layer1 XW	Layer2 XW	Max Diff
Origin Implemention Small/4096/4096/64/16/32/manual_time	4.08 ms	6.22 ms	168	3.02151	1.05938	0
Origin Implemention Small/4096/4096/128/16/32/manual_time	8.71 ms	11.0 ms	83	7.64268	1.07005	0
Origin Implemention Small/4096/16384/64/16/32/manual_time	4.31 ms	8.47 ms	160	3.25141	1.05807	0
Origin Implemention Small/4096/16384/128/16/32/manual_time	8.52 ms	12.8 ms	82	7.47336	1.04731	0
Origin Implemention Small/4096/65536/64/16/32/manual_time	4.09 ms	14.4 ms	171	3.0319	1.05445	0
Origin Implemention Small/4096/65536/128/16/32/manual_time	8.55 ms	19.0 ms	82	7.47175	1.08111	0
Origin Implemention Small/16384/16384/64/16/32/manual_time	17.7 ms	28.2 ms	40	12.5433	5.15441	0
Origin Implemention Small/16384/16384/128/16/32/manual_time	34.1 ms	42.9 ms	21	29.8766	4.20534	0
Origin Implemention Small/16384/65536/64/16/32/manual_time	16.3 ms	33.2 ms	43	12.0773	4.25599	0
Origin Implemention Small/16384/65536/128/16/32/manual_time	34.2 ms	51.6 ms	21	29.9769	4.20568	0
Origin Implemention Small/65536/65536/64/16/32/manual_time	65.5 ms	101 ms	10	48.3707	17.0828	0
Origin Implemention Small/65536/65536/128/16/32/manual_time	136 ms	183 ms	5	118.802	16.7372	0
OpenBlas Implemention Small/4096/4096/64/16/32/manual_time	13.0 ms	48.0 ms	69	6.7392	6.26925	0
OpenBlas Implemention Small/4096/4096/128/16/32/manual_time	12.7 ms	46.4 ms	41	7.03614	5.70807	0
OpenBlas Implemention Small/4096/16384/64/16/32/manual_time	12.8 ms	48.2 ms	44	6.48939	6.30919	0
OpenBlas Implemention Small/4096/16384/128/16/32/manual_time	13.9 ms	48.5 ms	64	7.33681	6.52734	0
OpenBlas Implemention Small/4096/65536/64/16/32/manual_time	13.6 ms	50.7 ms	44	7.5862	6.05131	0
OpenBlas Implemention Small/4096/65536/128/16/32/manual_time	13.6 ms	51.4 ms	55	7.62979	5.95073	0
OpenBlas Implemention Small/16384/16384/64/16/32/manual_time	13.4 ms	48.8 ms	54	6.98568	6.39486	0
OpenBlas Implemention Small/16384/16384/128/16/32/manual_time	13.1 ms	49.0 ms	56	7.08602	5.98694	0
OpenBlas Implemention Small/16384/65536/64/16/32/manual_time	12.3 ms	51.6 ms	54	7.15749	5.11889	0
OpenBlas Implemention Small/16384/65536/128/16/32/manual_time	12.4 ms	51.8 ms	55	7.35711	5.05959	0
OpenBlas Implemention Small/65536/65536/64/16/32/manual_time	13.2 ms	55.2 ms	56	7.89813	5.32773	0
OpenBlas Implemention Small/65536/65536/128/16/32/manual_time	11.6 ms	50.1 ms	51	6.06239	5.568	0
Origin Implemention Standard/500000/500000/128/16/32/iterations:3/manual_time	1087 ms	1540 ms	3	923.534	163.522	0
Origin Implemention Standard/500000/1000000/128/16/32/iterations:3/manual_time	1139 ms	1697 ms	3	975.412	163.477	0
Origin Implemention Standard/500000/5000000/128/16/32/iterations:3/manual_time	1069 ms	2423 ms	3	905.607	163.511	0
Origin Implemention Standard/1000000/1000000/128/16/32/iterations:3/manual_time	2220 ms	3089 ms	3	1.89312k	326.766	0
Origin Implemention Standard/1000000/5000000/128/16/32/iterations:3/manual_time	2175 ms	3999 ms	3	1.84763k	327.122	0
Origin Implemention Standard/5000000/5000000/128/16/32/iterations:3/manual_time	10830 ms	15619 ms	3	9.22385k	1.60616k	0
OpenBlas Implemention Standard/500000/500000/128/16/32/iterations:3/manual_time	23.4 ms	268 ms	3	16.7036	6.70492	0
OpenBlas Implemention Standard/500000/1000000/128/16/32/iterations:3/manual_time	18.8 ms	347 ms	3	13.2099	5.63813	0
OpenBlas Implemention Standard/500000/5000000/128/16/32/iterations:3/manual_time	18.7 ms	828 ms	3	13.2971	5.40398	0
OpenBlas Implemention Standard/1000000/1000000/128/16/32/iterations:3/manual_time	37.2 ms	484 ms	3	27.6131	9.55953	0
OpenBlas Implemention Standard/1000000/5000000/128/16/32/iterations:3/manual_time	30.7 ms	1013 ms	3	23.634	7.11464	0
OpenBlas Implemention Standard/5000000/5000000/128/16/32/iterations:3/manual_time	124 ms	2509 ms	3	89	34.937	0

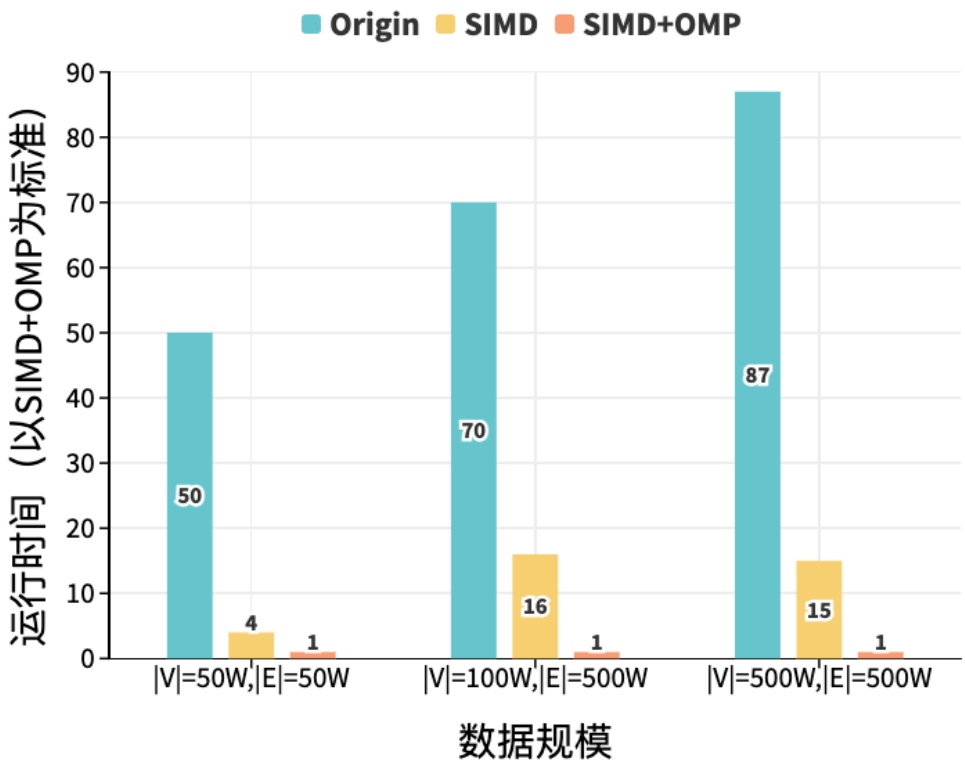
XW的最终效果

Benchmark	Time	CPU	Iterations	Layer1 AX	Layer2 AX	Max Diff
Origin Implemention Small/4096/4096/64/16/32/iterations:3/manual_time	1.13 ms	12.0 ms	3	0.313519	0.816804	0
Origin Implemention Small/4096/4096/128/16/32/iterations:3/manual_time	0.180 ms	11.6 ms	3	0.074616	0.105367	0
Origin Implemention Small/4096/16384/64/16/32/iterations:3/manual_time	0.502 ms	8.51 ms	3	0.181783	0.319739	0
Origin Implemention Small/4096/16384/128/16/32/iterations:3/manual_time	0.486 ms	13.0 ms	3	0.183142	0.302472	0
Origin Implemention Small/4096/65536/64/16/32/iterations:3/manual_time	1.64 ms	14.7 ms	3	0.594051	1.04834	0
Origin Implemention Small/4096/65536/128/16/32/iterations:3/manual_time	1.60 ms	19.7 ms	3	0.59477	1.00198	0
Origin Implemention Small/16384/16384/64/16/32/iterations:3/manual_time	2.13 ms	29.1 ms	3	0.630386	1.50184	0
Origin Implemention Small/16384/16384/128/16/32/iterations:3/manual_time	1.29 ms	47.3 ms	3	0.478595	0.81114	0
Origin Implemention Small/16384/65536/64/16/32/iterations:3/manual_time	2.06 ms	34.0 ms	3	0.811499	1.25166	0
Origin Implemention Small/16384/65536/128/16/32/iterations:3/manual_time	2.05 ms	54.4 ms	3	0.81372	1.24015	0
Origin Implemention Small/65536/65536/64/16/32/iterations:3/manual_time	5.22 ms	111 ms	3	1.47601	3.74838	0
Origin Implemention Small/65536/65536/128/16/32/iterations:3/manual_time	3.79 ms	187 ms	3	1.54663	2.24002	0
OpenBlas Implemention Small/4096/4096/64/16/32/iterations:3/manual_time	26.2 ms	39.9 ms	3	13.2228	12.9737	0
OpenBlas Implemention Small/4096/4096/128/16/32/iterations:3/manual_time	32.5 ms	39.0 ms	3	18.0223	14.4893	0
OpenBlas Implemention Small/4096/16384/64/16/32/iterations:3/manual_time	31.4 ms	47.8 ms	3	11.8948	19.4977	0
OpenBlas Implemention Small/4096/16384/128/16/32/iterations:3/manual_time	27.4 ms	51.7 ms	3	11.082	16.366	0
OpenBlas Implemention Small/4096/65536/64/16/32/iterations:3/manual_time	31.7 ms	45.0 ms	3	12.8085	18.8768	0
OpenBlas Implemention Small/4096/65536/128/16/32/iterations:3/manual_time	24.9 ms	45.6 ms	3	12.1698	12.7234	0
OpenBlas Implemention Small/16384/16384/64/16/32/iterations:3/manual_time	29.0 ms	44.8 ms	3	15.9084	13.0506	0
OpenBlas Implemention Small/16384/16384/128/16/32/iterations:3/manual_time	18.2 ms	39.1 ms	3	9.02429	9.17862	0
OpenBlas Implemention Small/16384/65536/64/16/32/iterations:3/manual_time	20.7 ms	40.4 ms	3	9.12316	11.5921	0
OpenBlas Implemention Small/16384/65536/128/16/32/iterations:3/manual_time	24.8 ms	47.6 ms	3	12.8311	11.9327	0
OpenBlas Implemention Small/65536/65536/64/16/32/iterations:3/manual_time	4.55 ms	37.2 ms	3	0.416485	4.13607	0
OpenBlas Implemention Small/65536/65536/128/16/32/iterations:3/manual_time	20.8 ms	54.2 ms	3	11.3058	9.52619	0
Origin Implemention Standard/500000/500000/128/16/32/iterations:3/manual_time	90.0 ms	1546 ms	3	24.9994	65.0441	0
Origin Implemention Standard/500000/1000000/128/16/32/iterations:3/manual_time	114 ms	1680 ms	3	30.2991	83.9206	0
Origin Implemention Standard/500000/5000000/128/16/32/iterations:3/manual_time	311 ms	2504 ms	3	102.219	208.717	0
Origin Implemention Standard/1000000/1000000/128/16/32/iterations:3/manual_time	176 ms	3155 ms	3	41.7713	134.221	0
Origin Implemention Standard/1000000/5000000/128/16/32/iterations:3/manual_time	433 ms	4078 ms	3	136.19	296.669	0
Origin Implemention Standard/5000000/5000000/128/16/32/iterations:3/manual_time	1179 ms	16269 ms	3	452.715	726.338	0
OpenBlas Implemention Standard/500000/500000/128/16/32/iterations:3/manual_time	31.1 ms	280 ms	3	6.94999	24.1868	0
OpenBlas Implemention Standard/500000/1000000/128/16/32/iterations:3/manual_time	17.7 ms	356 ms	3	3.32817	14.3372	0
OpenBlas Implemention Standard/500000/5000000/128/16/32/iterations:3/manual_time	57.7 ms	854 ms	3	15.7893	41.9339	0
OpenBlas Implemention Standard/1000000/1000000/128/16/32/iterations:3/manual_time	72.7 ms	482 ms	3	18.29	54.3879	0
OpenBlas Implemention Standard/1000000/5000000/128/16/32/iterations:3/manual_time	122 ms	1038 ms	3	31.1266	91.2133	0
OpenBlas Implemention Standard/5000000/5000000/128/16/32/iterations:3/manual_time	230 ms	2523 ms	3	97.7722	132.396	0

AX的最终效果

从上图可以看到

- 对于小数据集，我们优化效果一般就在3-4倍之间，这主要是由于数据集过小，并行的收益不是很大
- 对于较大规模的数据集，我们可以在不同的数据集上与原算法相比达到50-80倍的效果提升，优化效果十分的明显，较为充分的利用了CPU的性能。



从上图可以看到，我们在不同规模上进行测试，达到一个较好的效果，并且使用SIMD+OMP的优化，充分的利用硬件资源进行优化，具有可结合性。

总结

参加这个图计算挑战赛是我们团队的一次难忘经历，这个挑战让我们收获颇多。

- 我们的任务是对图卷积神经网络推理问题在CPU上进行计算优化，通过运用SIMD优化、快速矩阵相乘OpenBLAS库和OMP并行库等技术，我们成功地对推理过程进行了优化，从而在性能方面取得了显著的提升。

1. 首先，我们深刻了解了图卷积神经网络的推理过程。这个过程对于理解图计算的本质和复杂性至关重要。我们认识到了在CPU上面对大规模图数据的推理过程，所面临的挑战和瓶颈，这让我们明白了优化的重要性和紧迫性。
2. 其次，我们学到了如何运用SIMD优化来提高图计算的性能。SIMD指令集的并行计算特性让我们能够同时处理多个数据元素，这为图计算的加速提供了有力的支持。我们对代码进行了重构和优化，使得CPU能够更高效地并行处理图数据，从而在推理过程中显著提升了性能。
3. 在优化过程中，我们发现了快速矩阵相乘OpenBLAS库的潜力。OpenBLAS是一个高性能的数学库，其优化了矩阵乘法运算，能够在CPU上高效地进行大规模的矩阵计算。将图计算转化为矩阵计算，并结合OpenBLAS的使用，使得推理过程的计算复杂度降低，性能得到了进一步的提升。
4. 同时，我们深入学习了OMP并行库的使用。OMP是一种并行编程框架，它能够简化多线程编程的过程，充分利用多核CPU的计算能力。通过在关键的计算部分加入OMP并行化，我们有效地提高了推理过程的并行性，让CPU资源得到更充分的利用，从而进一步加速了推理过程。
5. 在挑战中，我们更深刻地体会到了团队合作的力量。合作是取得优异成绩的关键，通过团队成员之间的相互支持、分工合作以及不断交流和讨论，我们共同攻克了一个个难题，最终取得了优秀的成绩。团队合作不仅提高了效率，还为我们带来了更多的乐趣和成就感。

这次挑战让我们对图计算和优化有了更深刻的认识和理解。我们不仅掌握了新的技术和工具，也提升了解决问题的能力和创新思维。这将对我们未来的学习和职业发展产生重要的影响。我们会继续保持学习的热情，不断挑战自我，迎接更多的技术挑战，并期待能在图计算领域取得更为卓越的成就。