

Adam Shirley

Project: K-Means clustering of Weather data

Description:

In the project i will use K-means clustering to cluster weather data from Noah of the Los angeles area using the high temperature off each day as the x coordinate and the low as the y. I will then print the number of days of each month in each cluster to observe if it is possible to spot dependencies on high and low temperatures by month.

Instructions:

To run program compile main.cu with nvcc compiler followed by calling the executable on the command line with 2 parameters, the first being the number of clusters or k, and the second being the name of the files where the input data is located. The code will output a list of the data by cluster to output.csv as well as print a table to both the command line and file of a table of the cluster and number of points of a specific month in each cluster

Pseudo Code:

```
main()
    readFromFileToArrayofStructureDay()
    copyDataToDevice()
    while clusters remain unchanged:
        cluster<<<numDays/512, 512>>>(data)//done on device
        calculaeNewCenters<<<k, 32>>>()//done on Device
    printDataToOutput.csv()
    array months[k*12]//array for number of data points from each month in each cluster
    copy(months to device);
    countMounths<<<k, (12, 32)>>>(months);
    printMonths
```

```
__global__ void cluster()
    int i=0
    int index;
    int dist = MAX;
    while((index=threadIdx.x+gidDim.x*i++)<numDays)
        for each cluster:
            calculate Distance To center;
            if dist<min
                data.cluster=cluster
```

```

__global__ void calculate_centers()
{
    __shared__ int nums[3];
    int * avgx =nums;
    int * avgy=&nums[1];
    int * n=&nums[2];
    int i=0;
    int index;
    while((index=threadIdx.x+blockDim.x*i) < numDays){
        if data[index].cluster == blockIdx.x){
            atomicadd(n,1)
            atomicAdd(avgx, data[index].high)
            atomicAdd(avgy, data[index].low)

            i++
        }
    }
    __syncthreads()
    if(threadIdx.x ==0)
        centers[blockIdx.x].x=avgx/n
        centers[blockIdx.x].y=avgy/n
}

```