Hierarchical clustering

1 Description

Hierarchical clustering is commonly used in exploratory data analysis. It is used to determine clusters of similar data points in multidimensional spaces.

At the beginning of hierarchical clustering, each data point is considered a separate cluster. The two clusters that are closest according to some metric are joined to form a new cluster. This process is repeated till all of the points belong to one cluster. The final hierarchical cluster structure is called a dendrogram.

Two kinds of distance metrics are employed in the process.

1. The first kind determines the distances between two individual data points and will be referred to as pairwise distances. Some common choices are Euclidean distance, Manhattan distance, and Pearson correlation coefficient.
2. The second kind of distance metrics concerns the distances between clusters consisting of multiple data points.

Let us assume that the data are an $n \times m$ matrix M, stored in a float array called in in row-major order. We need to compute the pairwise distances. A sequential code to compute this pairwise distance is

```
void cpuRho(float *out, float *in, int n, int m){
  int i, j, k;
  float x, y, a1, a2, a3, a4, a5;
  float avgX, avgY, varX, varY, cov, rho;
  for(i=0;i<n;i++){
    out[i*n + i] = 1.0;
    for(j=i+1;j<n;j++){
      a1 = a2 = a3 = a4 = a5 = 0.0;
      for(k=0;k<m;k++){
        x = in[i*m+k], y = in[j*m+k];
        a1 += x, a2 += y;
        a3 += x*x, a4 += y*y, a5 += x*y;
      }
      avgX = a1/m, avgY = a2/m;
      varX = (a3 - avgX*avgX*m)/(m-1);
      varY = (a4 - avgY*avgY*m)/(m-1);
      cov = (a5 - avgX*avgY*m)/(m-1);
      rho = cov/sqrtf(varX*varY);
      out[i*n + j] = rho;
    }
  }
}
```

Pearson correlation coefficient ρ between rows i and j is defined as

$$\rho(i,j) = \frac{1}{m-1} \sum_{k=1}^{m} \left(\frac{M_{ik} - avg(i)}{std(i)} \right) \left(\frac{M_{jk} - avg(j)}{std(j)} \right)$$

2 Questions

In this set of question, we assume that the GPU has p compute unit and the input array is of size $n \times m$.

2.1 Design

In the first part, we consider the design aspect of the algorithm in parallel.

- Perform the first stage of the Foster methodology (decomposition).
  - What is the granularity of your decomposition ?
  - How did you select it ?
  - What form of decomposition did you select ?
- Perform the second stage of the Foster methodology (communication).
  - How many exchanges are required ?
  - What is the volume of data to be exchange ?
  - Does it depends on the number of data ?
- Perform the third stage of the Foster methodology (grouping).
  - How did you decide the size of the grouping ?
  - Does the grouping depend on the number od data ?

2.2 GPU consideration

We assume that we have a classical GPU organization of the p cores.

- How will you organize the computation on p compute units organized in warp ?
- What constraints do you need to consider ?