Parallel Spectral Graph Partitioning on CUDA

Orhan Firat¹, Alptekin Temizel²
1. Department of Computer Engineering, Middle East Technical University
2. Graduate School of Informatics, Middle East Technical University

Spectral Clustering
Spectral clustering is a graph-theoretic clustering algorithm
Weighted Graph G=(V, E, A)
min between-cluster similarities (weights : A)

Algorithm flow
Each blue box represents a module applied on GPU
Green box represents the CPU step

Boosting steps
Distance matrix
Calculate distance from each point to others
\[ d(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \cdots + (z_i - z_j)^2} \]

Find t-nearest neighbor
Extract closest points row-wise from its diagonal
Sort rows of the distance matrix
1. Insertion sort with kernel
2. Sequential sort with thrust::sort

Generate Laplacian
Compute diagonal matrix D as the row sum
Using CUBLAS cublas_dgemv
Multiply with Similarity matrix on left/right Using CUBLAS cublas_dgemm
Launch 1 additional kernel to symmetrize

CPU vs GPU

Overall Run Time
CPU Configuration 1:
Step 1.1: thrust::sort_by_key
Step 1.2: memory operations (host to device)
Step 1.3: memory operations (device to host)
Step 1.4: memory operations (device to host)
Step 1.5: memory operations (device to host)
Step 1.6: kernel executions
Step 1.7: memory operations (host to device)
Step 1.8: memory operations (host to device)
Step 1.9: memory operations (host to device)
Step 1.10: memory operations (device to host)
Step 1.11: memory operations (device to host)
Step 1.12: memory operations (host to device)

CPU Configuration 2:
Step 1.1: cuda insertion sort
Step 1.2: memory operations (host to device)
Step 1.3: memory operations (host to device)
Step 1.4: memory operations (device to host)
Step 1.5: memory operations (host to device)
Step 1.6: memory operations (device to host)
Step 1.7: memory operations (host to device)
Step 1.8: memory operations (device to host)
Step 1.9: memory operations (host to device)
Step 1.10: memory operations (device to host)
Step 1.11: memory operations (device to host)
Step 1.12: memory operations (host to device)

Overall Running Time

Conclusion
- CUBLAS library is extensively used in this study and found very useful in matrix routines which are the main axis of spectral clustering
- Calculation of t-nearest neighbors in a large dense distance matrix is the most challenging part of spectral clustering
- CUDA-Thrust library is also a very useful tool when dealing with array-typed data structures.

References