Accelerating Gaussian Mixture Model Training with Parallel EM on GPUs

Clint Zhu (clintz)
Chenghui Yu (cyu4)

https://seris370.github.io/15618-project-web/

Schedule:

Our original schedule remains largely intact. Based on our progress and a better understanding of implementation complexity, we have refined the remaining plan into half-week increments with assigned responsibilities.

Title	Date	Status
Setup data loading and GMM pipeline	Nov 22, 2…	Completed
CPU baseline EM implementation	Nov 24, 2…	Completed
Parallel CUDA E-step kernel	Nov 24, 2…	Completed
Test code to validate cuda E-step	Nov 24, 2…	Completed
Parallel CUDA M-step kernel	Dec 2, 2025	In progr…
Test code to validate cuda M-step	Dec 2, 2025	Not started
Benchmarking baseline EM	Dec 4, 2025	Not started
Experimenting with clustering datasets	Dec 6, 2025	Not started
Final report	Dec 8, 2025	Not started

So far, we have successfully implemented the full GMM training pipeline connecting data loading, initialization, and iterative EM training in Python. A sequential CPU-only EM implementation has been completed and thoroughly validated against small mock datasets to ensure correctness. This baseline includes full support for log-likelihood computation, responsibility normalization, and parameter updates.

On the GPU side, we have implemented a CUDA-accelerated parallel E-step, which computes log-probabilities and responsibilities for each data point–component pair. The kernel maps each

(N × K) pair to GPU threads and leverages batched memory access for feature vectors and Gaussian parameters. This kernel has been fully integrated into the Python training loop and produces numerically consistent results with the CPU baseline.

We also completed the end-to-end pipeline, including dataset loading, train-test split, and EM iteration control, allowing us to switch seamlessly between CPU and GPU E-step execution.

Adjusted expectations

Advanced warp-level and shared-memory optimizations will be attempted but are not guaranteed.

Poster (Final Report) Goal

- Performance graphs showing E-step and full EM speedups
- Scalability plots with respect to N, K, and feature dimension D
- Architecture diagrams illustrating the CUDA kernel design
- A demo that:
    - Loads a dataset
    - Trains the GMM using GPU EM
    - Displays clustering results for 2D data if applicable

Preliminary Results

E-step results versus baseline

N (sample)	D (dim)	K (cluster)	CPU	GPU	Speedup
1000	2	3	0.00316s	0.00202s	1.56x
1000000	2	3	0.213s	0.025s	8.52x
10000000	2	3	2.01s	0.25s	8.04x
20000000	2	3	4.049s	0.515s	7.86x

Initial results already show clear acceleration of the E-step on the GPU, especially as N increases, confirming the effectiveness of CUDA parallelization.

Conclusion

Overall, the project is on schedule and progressing as proposed. The successful integration of the CPU baseline, full pipeline, and CUDA-based E-step confirms the technical feasibility of our approach. With remaining work focused on the parallel M-step and benchmarking, we are

confident that we will achieve our minimum deliverables and several stretch goals before the final submission.