1. Overview

- Develop new techniques and approaches to speed up machine learning algorithms for large-scale data processing.
- Objectives
  - Explore parallel implementations of machine learning algorithms on computer clusters with and without a variety of accelerators
  - Develop new capacity planning models to give a data analyst quick information about the execution time of a machine learning algorithm
- Driven applications: the classic k-means clustering algorithm parallelized using MATLAB Parallel Computing Toolbox.

2. Parallelization

- Three general classes of unsupervised learning methods:
  - K-means clustering
  - Mixture models
  - Latent Dirichlet Allocation (LDA)
- K-means clustering
  - Parallelize the "assignment" in each iteration
  - MATLAB Parallel Computing Toolbox, parfor, spmd
  - CUDA or MPI programming

3. Case Study

- Parallel k-means clustering for image processing

4. Performance Prediction

- FiM – a Fine-grained Markov model for predicting execution times
  - state of each thread: active ($R_i^0$) or passive ($R_i^1$)
  - partition the processing of an application into multiple stages
  - Transition probabilities: switch between two states ($P_{ij}$) or self-loop in the same state ($P_{ii}$)
  - Completion probabilities: transfer from one stage to the other one (e.g., $P_{s1}$ and $P_{s2}$)
  - Instrumentation tools (perf stat) to obtain required data
  - Execution time per stage: $T_i = 1/(P_{si} \cdot F)$
- Learning trend to refine models to be independent on hardware parameters
  - develop a new regression model with a multiplicative tuning factor

5. Preliminary Results

- (a) Learning trend of total cycles, stalled cycles and utilizations
- (b) Predicted results using the modified linear regression model
- (c) Actual v.s. Predicted results

Summary of this project

- Capture the characteristics of heterogeneous hardware architectures.
- Predict the behavior of an application running on an array of computing platforms.
- Configure the right computing platform and choose a good way (MATLAB Parallel Computing Toolbox, CUDA or MPI) to parallelize unsupervised learning algorithms.

Drivne applications: the classic k-means clustering algorithm parallelized using MATLAB Parallel Computing Toolbox.