CS 359 Parallel Computing Project Proposal

Parallel Implementation of the Conjugate Gradient Method

Under Guidance of
Dr. Surya Prakash

By:

Aditi Ganvir - cse210001016
Princy Sondarva - cse210001068

Introduction:

The Conjugate Gradient Method (CGM) is an iterative numerical technique for solving systems of linear equations, with a particular focus on large sparse symmetric positive-definite linear systems, especially when dealing with large, sparse matrices. It typically converges much faster than direct methods (e.g., Gaussian elimination) for such systems. Moreover, it does not require the storage and factorization of the entire matrix, making it suitable for problems with limited memory resources. This method is also used in solving various unconstrained optimization problems like energy minimization.

It's super efficient when working with sparse matrices, which are matrices with many zeros. Instead of caring about the matrix's overall size, it mainly depends on the number of non-zero values. This is a big advantage when you have limited memory and processing power.

In this project, our objective is to develop a parallel version of the Conjugate Gradient Method, which can be applied to multiple processors or computing resources to improve efficiency.

Methodology:

Since this is an iterative method, the result at the next step depends on the result at the current step. Thus, it is not possible to simultaneously compute the intermediate steps. However, we can parallelize the operations in a step.

Here's a step-by-step overview of how the Conjugate Gradient Method works(taken from wikipedia):

Initialize the solution vector x and the initial residual r0, where r0 = b - A * x0.

Compute the first search direction p0, which is equal to the initial residual: p0 = r0.

For each iteration (k = 0, 1, 2, ...), do the following:

a. Calculate the step size (or scaling factor) αk by minimizing the residual along the chosen direction:

αk = (r_k^T * r_k) / (p_k^T * A * p_k)

b. Update the solution vector:

x_(k+1) = x_k + αk * p_k

c. Update the residual:

r_(k+1) = r_k - αk * A * p_k

d. Calculate the parameter βk to ensure that the search directions are conjugate:

βk = (r_(k+1)^T * r_(k+1)) / (r_k^T * r_k)

e. Update the search direction:

p_(k+1) = r_(k+1) + βk * p_k

Repeat the iterations until a stopping criterion is met. Common stopping criteria include reaching a specified tolerance, a maximum number of iterations, or other convergence criteria.

References:

- Wikipedia. Conjugate gradient method. (https://en.wikipedia.org/wiki/Conjugate_gradient_methd)