Drexel University

Electrical and Computer Engineering Dept.

ECEC-413

Assignment 3:

Pthread Jacobi Solver

Minjae Park

John Truong

Professor Naga Kandasamy

Date: 05/06/2020

Jacobi Solver using Barriers:

Pthread implementation of Jacobi Solver utilized barriers for synchronization between iterations. The barrier was initialized in the main thread, and pthread_barrier_wait was called by the created threads for synching. Since the convergence factor needed to be calculated for the entire matrix, a mutex lock was declared for protected writing between threads. There are 2 fork points in this program: Initialization of the X matrix and the Jacobi Iteration for updating the X matrix. The initialization step, which copies matrix B to X, was parallelized by creating threads to the function jacobi_setup. After all the threads join the main thread, thread arguments are updated with the global ssd variable, barrier and mutex addresses and threads are created for Jacobi iteration. The created threads run through the iteration of calculating portions of the X matrix. Once the threads exit, the main thread calculates the convergence factor mse and checks whether the program should exit the while loop. At the end of the main function, data structures are freed before exiting.

Results:Speed Up:

Matrix Size	Single Thread	4 Threads	8 Threads	16 Threads	32 Threads
512x512	3.717	3.432	6.672	16.037	26.452
1024x1024	29.363	12.135	20.578	29.013	53.581
2048x2048	251.689	79.362	69.894	119.535	108.522

Table 1: Jacobi Solver Data on Xunil-03

Matrix Size	Single Thread	4 Threads	8 Threads	16 Threads	32 Threads
512x512		108.30%	55.71%	23.18%	14.05%
1024x1024		241.97%	142.69%	101.21%	54.80%
2048x2048		317.14%	360.10%	210.56%	231.92%

Table 1: Speedup of Jacobi Solver