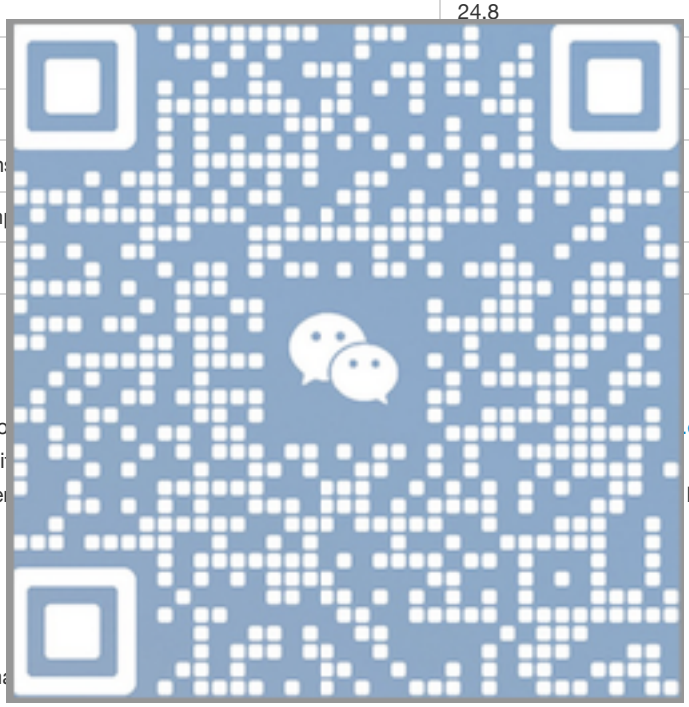
COSC3500/7502 Assignment: Parallel programming techniques

1. Summary

The goal is to implement three different matrix multiply functions for three different hardware configurations (CPU - AVX/openMP, GPU - CUDA, and a cluster of two nodes - MPI). The matrices are real-only square matrices. Performance will be benchmarked relative to a reference implementation (Intel MKL for CPU and MPI, CUBLAS for GPU) for a 2048×2048 matrix, and marks will be assigned based on speed.

2. Rubric

Marks	CPU (AVX/openMP) 4 cores	GPU (CUDA) 1GPU	MPI 2 nodes, 4 cores each
7	2.3	4.9	1.4
6	16.5	8	8.3
5	36.2	24.8	18.5
4	64.3		31.6
3	121		60.8
2	1000 (and gives correct an		
1	Compiles and runs to com		
0	Doesn't compile or wasn't		



3. Hardware

- Final performance will be assessed c .edu.au cluster.
- For development, jobs can be submi
- All nodes on rangpur have similar pe highly optimized MPI matrix multiply will show communication overhead.

4. The benchmarks

- Random unitary square real - only m and speed relative to reference implementations.
- For CPU and GPU, matrix multiplication is on the same machine. For MPI, each node has its own copy of matrices from the start, and nodes need to maintain a copy of the current matrix product answer.

5. Turning on/off CPU, GPU, and/or MPI code

- Comment out the relevant `#define` lines in `Assignment1_Gradebot.cpp` and remove relevant lines in the `MakeFile` if the corresponding hardware is not available.

6. The only code files you can modify for your final submission

- Only 3 files can be changed: `matrixMultiply.cpp` , `matrixMultiplyGPU.cu` , and `matrixMultiplyMPI.cpp` .
- Functions must not use outside libraries (except provided headers) and must not write to stdout or file in the final submission.

7. The script for final grade and debugging

- The script for final grade is `goslurm_COSC3500Assignment_RangpurJudgementDay` .

- For debugging, use variations of `goslurm_COSC3500Assignment_RangpurDebug` or `goslurm_COSC3500Assignment_GetafixDebug` scripts.

8. Software interface and the GradeBot

- `Assignment1_GradeBot.cpp` runs benchmarks and assigns marks.
- Usage:
  `./Assignment1_GradeBot {matrix dimension} {threadCount} {runBenchmarkCPU} {runBenchmarkGPU} {runBenchmarkMPI} {optional integer} {`

9. Text output

- The `Assignment1_GradeBot` outputs to stdout and individual text files for each benchmark on each node (`COSC3500Assignment_{benchmark type}_{node}.txt`).
- The text files include 6 columns: Info., N, Matrices/second (MKL), Matrices/second (You), Error, Grade.

10. Final Submission

- Submission must include `matrixMultiply.cpp`, `matrixMultiplyGPU.cu`, `matrixMultiplyMPI.cpp` and a zip file `slurm.zip` (containing slurm job output files) all zipped together in a file named `{your 8 digit student number}.zip`. If a required file is not implemented, submit the original blank file.