



## CodeCrunch

### CS2030 Matrix Multiplication

#### Tags & Categories

Tags:

Categories:

#### Related Tutorials

#### Task Content

##### Matrix Multiplication

##### Problem Description

Matrix multiplication is a fundamental operation with many applications in physics, engineering, mathematics, and computer science.

Given a matrix  $A_{n \times m}$  of  $n$  rows by  $m$  columns, and a matrix  $B_{m \times p}$ , the matrix product  $C_{n \times p} = AB$  is an has elements  $c_{ij}$  given by

$$c_{ij} = \sum_{k=1}^m m_k = a_{ik} b_{kj}$$

In this lab, we are interested in parallelizing the following divide-and-conquer algorithm for matrix multiplication. Let

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

where  $A_{11}$ ,  $A_{12}$ , etc. are block partitioned matrices of **equal** sizes. If  $C = AB$ , then:

$$\begin{aligned}
 C &= \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \\
 &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \\
 &= \begin{bmatrix} A_{11}B_{11}+A_{12}B_{21} & A_{11}B_{12}+A_{12}B_{22} \\ A_{21}B_{11}+A_{22}B_{21} & A_{21}B_{12}+A_{22}B_{22} \end{bmatrix}
 \end{aligned}$$

### The Task

You are to implement the above divide-and-conquer algorithm as a `RecursiveTask` and submit it to `ForkJoinPool` for execution. For simplicity, we only need to handle square matrices of size  $2^n$  for  $n$  up to 11.

A skeleton file [MatrixMultiplication.java](#) has been provided for you. The class `MatrixMultiplication` inherits from `RecursiveTask`, with the necessary fields and constructor. Your task is to complete the `compute` method.

The file [Matrix.java](#) is also provided for you. It implements a matrix with double values, and stores the values of the matrix in a 2D double array `m`. It also stores the dimensions of the matrix in the field `dimension`. It includes two methods to multiply two matrices, one sequentially with triple for loops, and another (also sequentially) with the recursive divide-and-conquer algorithms. There is a method to compare if two matrices are equal.

In addition, the method `parallelMultiply` invokes the parallel version of matrix multiplication. At this moment, the method simply calls the non-parallel version of `recursiveMultiply`.

The file [Main.java](#) provides the driver class which measures the timing runs for a given matrix multiplication task three times before taking the average.

Points to note:

- Find a suitable `FORK_THRESHOLD` for `MatrixMultiplication` such that any matrix dimension smaller than this threshold would be better off using sequential matrix multiplication.
- Try with small matrices first. Make sure the code is correct before you go for larger matrices.
- You should not spawn too many tasks that block, which will in turn lead to too many compensation threads being created in `ForkJoinPool`, and a `RejectedExecutionException` being thrown.
- For matrices of dimensions  $2^{10}$  and  $2^{11}$ , you need to run `java` with the argument `-Xmx[size]` to increase the heap memory size. For example, `-Xmx1g` increases the heap memory up to 1GB, and should work well for both cases. That said, you should still not create too many unnecessary copies of the matrices.
- If you grow impatient while waiting and want to stop the running process, type `Control-C` in your `ssh` window. You may have to wait up to a few seconds for the process to stop.
- You need to only submit `Matrix.java` and `MatrixMultiplication.java` to CodeCrunch.

### Submission (Course)

Select course:

CS2030 (2018/2019 Sem 2) - Programming Methodology II ▼

Your Files:

BROWSE

SUBMIT (only .java, .c, .cpp, .h, and .py extensions allowed)

To submit multiple files, click on the Browse button, then select one or more files. The selected file(s) will be added to the upload queue. You can repeat this step to add more files. Check that you have all the files needed for your submission. Then click on the Submit button to upload your submission.