ECEN424 Fundamentals of Networking MP 2 - Due Date: See lecture01

Java Threads

Created by: Dr. Srinivas Shakkottai *Scribe: Self*

1 Introduction

In this machine problem, you will learn about concurrency and multithreaded programming in Java. Threads are lightweight processes, and the creation of a new thread requires less system resources as compared to the creation of a process. Multithreaded programming allows multiple threads to exist within the context of a single process. These threads execute independently, sharing the process' resources. A multithreaded program operates faster on machines having multiple cores.

2 Problem Statement

In this assignment, you will implement a multithreaded version of matrix multiplication. The normal method for multiplying two such matrices involves performing all the calculations in the main thread. Each element (i, j) of the product matrix is obtained by multiplying the i^{th} row of the first matrix with the j^{th} column of the second. In the multithreaded version, you will divide this computation among threads to achieve parallelism. Initialize two 20×20 matrices with random values. Create five threads. Each thread will compute $1/5^{th}$ of the product matrix. So the first thread calculates the product elements for the first four rows, the second for the next four rows and so on. The main thread should wait for all the threads to complete, and then print the resultant product matrix. Verify that this produces the same result as the normal method.

3 Points to Note

Create the matrices outside the logic for the threads. Call "join" to ensure that the main threads prints the product only after all computations are complete.

4 Evaluation Guidelines

1. Check the operation using by verifying using an online matrix multiplier (eg: `http://wims.unice.fr/wims/en_tool~linear~matmult.en.html`).

2. Check the code to verify that multithreading is being used.

References

[1] http://docs.oracle.com/javase/tutorial/essential/concurrency/index.html