3.2 Lecture Summary

3 Loop Parallelism

3.2 Parallel Matrix Multiplication

In this lecture, we reminded ourselves of the formula for multiplying two $n \times n$ matrices, a and b, to obtain a product matrix, c, of size $n \times n$:

$c[\,i\,][\,j\,] = \sum_{k=0}^{n-1} a[\,i\,][\,k\,] * b[\,k\,][\,j\,]$

This formula can be easily translated to a simple sequential algorithm for matrix multiplication as follows (with pseudocode for counted-for loops):

```
1  for(i : [0:n-1]) {
2    for(j : [0:n-1]) { c[i][j] = 0;
3      for(k : [0:n-1]) {
4        c[i][j] = c[i][j] + a[i][k]*b[k][j]
5      }
6    }
7  }
```

The interesting question now is: which of the for-i, for-j and for-k loops can be converted to forall loops, i.e., can be executed in parallel? Upon a close inspection, we can see that it is safe to convert for-i and for-j into *forall* loops, but for-k must remain a sequential loop to avoid data races. There are some trickier ways to also exploit parallelism in the for-k loop, but they rely on the observation that summation is algebraically associative even though it is computationally non-associative.

Optional Reading:

1. Wikipedia article on Matrix multiplication algorithm

Mark as completed