CS614: Advanced Compilers

Fall 2023 (Due: November 6th, 2023)

Assignment A4: Can you Parallelize me?

---

1 Assignment Objective

Use JavaCC and JTB to perform loop parallelization.

2 Detailed Specification

Parallelization enables a loop to run over multiple processors (or threads) with a potentially significant execution speedup. Beware that not all loops are parallelizable. Running a loop in parallel over several processors can result in iterations executing out of order. Moreover, the multiple threads executing the loop in parallel may interfere with each other whenever there are data dependencies in the loop.

Our task is to iterate over the body of methods and perform loop parallelization based on simple alias and dependence analysis for references and array indices. We will use GCD test as discussed in the class.

Additional Information:

- Classes will not contain any fields.
- Assume that objects coming as parameters can point to any memory location.

3 Public Testcase

3.1 Input

```
class Test {
    public static void main(String[] arg) {
        Operator operator;
        int[] arr1;
        int[] arr2;
        int[] res;
        operator = new Operator();
        arr1 = new int[250];
        arr2 = new int[250];
        res = operator.operate(arr1,arr2);
    }
}
```

```
class Operator {
    public int[] operate(int[] arr1, int[] arr2) {
        int[] arr3;
        int[] arr4;
        int i;
        int j;
        int k;
        arr3 = new int[100];
        arr4 = new int[1000];
        //Trivially Parallelizable loop.
        for(i = 1; i <= 200; i=i+1){
            arr4[i] = i;
        }
        //Parallelizable loop using GCD Test
        for(i = 1; i <= 10; i=i+1){
            k = i * 2;
            j = i * 10;
            j = j + 51;
            arr3[i] = arr4[k];
            arr4[j] = k;
        }
        //Non-Parallelizable loop using GCD Test.
        for(i = 0; i < 200; i=i+1){
            j = 200 - i;
            j = j - 1;
            arr4[i] = arr4[j];
        }
        arr3 = arr2;
        arr4 = arr1;
        //Non-Parallelizable loop. (arr3 and arr4 can be aliases)
        for(i=0; i<99; i=i+1){
            j = i + 1;
            arr3[i] = arr4[j];
        }
        return arr4;
    } }
```

3.2 Output

```
class Test {
    public static void main(String[] arg) {
        Operator operator;
        int[] arr1;
        int[] arr2;
        int[] res;
        operator = new Operator();
        arr1 = new int[250];
        arr2 = new int[250];
        res = operator.operate(arr1,arr2);
    } }

class Operator {
    public int[] operate(int[] arr1, int[] arr2) {
        int[] arr3;
        int[] arr4;
        int i;
        int j;
        int k;
        arr3 = new int[100];
        arr4 = new int[1000];
        for(/* @Parallel */i = 1; i <= 200; i=i+1){
            arr4[i] = i;
        }
        for(/* @Parallel */i = 1; i <= 10; i=i+1){
            k = i * 2;
            j = i * 10;
            j = j + 51;
            arr3[i] = arr4[k];
            arr4[j] = k;
        }
        for(i = 0; i < 200; i=i+1){
            j = 200 - i;
            j = j - 1;
            arr4[i] = arr4[j];
        }
        arr3 = arr2;
        arr4 = arr1;
        for(i=0; i<99; i=i+1){
            j = i + 1;
            arr3[i] = arr4[j];
        }
        return arr4;
    } }
```

4 Evaluation

Your submission must be named as `rollnum-a4.zip`, where `rollnum` is your roll-number in small letters. Upon unzipping the submission, we should get a directory named `rollnum-a4`. The main class inside this directory should be named `Main.java`. Your program should read from the standard input and print to the standard output. You can leave all the visitors and syntax-tree nodes as it is, but remember to remove all the `.class` files and `.jar` files.

We would run the following commands as part of the automated evaluation process:

- `javac Main.java`
- `java Main < test > out`

If the contents of `out` file match with the expected output for the testcase, you would get marks for the corresponding testcase. Make sure to verify using the `CompL Evaluator` before submission.

5 Plagiarism Warning

You are allowed to discuss publicly on class and slack, but are supposed to do the assignment completely individually. We would be using sophisticated plagiarism checkers, and if similarity is found, the penalty used in the course would be as follows:

- First instance: 0 marks in the assignment
- Second instance: FR grade.
- Third instance: Report to institute disciplinary committee.

-*-*-*- Do the assignment honestly; enjoy learning the course. -*-*-*-