

FOR & PARALLEL-FOR Examples

Yehong Wang
11/16/2009

What CAN be inside PARALLEL-FOR:

- creation of scalar variables
- deletion of variables
- matrix/vector indexing
- non-matrix-operation assignments
- if-else statements with only <What CAN be inside parallel-for>
- for statement with only <What CAN be inside parallel-for>
- synchronization statements
- comments

What CANNOT be inside PARALLEL-FOR:

- creation of matrix/vector/structure variables
- matrix operations (multiplication, addition, subtraction, transpose)
- if-else statements with only <What CANNOT be inside parallel-for>
- for statement with only <What CANNOT be inside parallel-for>
- parallel-for statements
- structure definitions
- function definitions
- function invocations
- print statements
- plot commands

What CAN be inside FOR

- creation of ANY variables
- deletion of variables
- matrix/vector indexing
- function invocations
- ANY assignment (including matrix/vector operations)
- FOR/PARALLEL-FOR statements
- if-else statements;
- plot commands
- comments

What CANNOT be inside FOR

- print statements
- structure definitions

function definitions
synchronization statements

Examples

// 1. Matrix indexing inside PARALLEL-FOR

```
Matrix<int>[2][100] c = [ ];
parallelfor(x->0 to 99)
{
    c[1][x] = a[1][x] + b[1][x];
}
```

// 2. Vector indexing & if-else statement inside PARALLEL-FOR

```
Vector<int>[100] z = [ ];
parallelfor(x -> 0 to 99)
{
    int y;
    y = 0;
    if ( y == 0) {
        z[x] = x + x;
    }
    else {
        z[x] = x * x;
    }
}
```

// 3. FOR statement inside PARALLEL-FOR

```
Vector<int>[10] z = [ ];
parallelfor(x -> 0 to 9)
{
    z[x] = x;
    for (y -> 1 to 3)
    {
        z[x] = z[x] * z[x];
    }
}
```

// 4. Synchronization statement and comments inside PARALLEL-FOR

```
Matrix<int>[2][100] c = [ ];
parallelfor(x->0 to 99)
{
    // this is comment
    c[0][x] = x * (2 + x);
    sync;
    c[1][x] = c[0][x] + 1;
}
```