Explicit Vector Code

ACTIVITY 4-4: USING CILK PLUS ARRAY NOTATION

In this activity you will implement explicit vectorisation by

- using array notation
- using elemental functions

Using Array Notation

1. Edit the file `chapter4.c` and change the matrix multiplication function to use array notation in the inner most loop. The code that has to be added is enclosed in the box

```
void MatrixMul( double a[ N][ N], double b[ N][ N], double c[ N][ N])
{
  int i, j;
  for (i = 0; i < N; i++)
  {
    for (j = 0; j < N; j++)
    {
      #ifdef USE_ARRAY_NOTATION
        c[ i][ j] = __sec_reduce_add( a[ i][:] * b[:][j]);
      #else
            for (k=0; k<N; k++) {
              c[i][j] += a[i][k] * b[k][j];
            }

      #endif
    }
  }
}
```

Questions about the code

a. What is the purpose of the `[:]` syntax?
b. Why has the variable 'k' not been used in the array notation code?
c. What does the `__sec_reduce_add` function do?

2. Build the application:

- Linux

```
make clean
make CFLAG = "-O2 –DUSE_ARRAY_NOTATION" TARGET=intel.cean
```

- Windows

```
nmake clean
nmake CFLAGS= " /O2 –DUSE_ARRAY_NOTATION" TARGET=intel.cean
```

3. Run the program `intel.cean.exe`.

If you are keen, you can check the results are the same as the original code. ☺

4. Rebuild the code adding the option -S, and look at the generated assembler file.

Questions (b and c are optional)

a. *Does the code look vectorised?*

b. *Compare the assembler from this version with the code generated in the previous lab. Do they look similar?*

c. *Does the code have CPU dispatch (this is a hard question), if not how do you suppose you could enable CPU dispatch on array notation (you may have to experiment).*

ACTIVITY 4-5: USING AN ELEMENTAL FUNCTION

In this activity you will create an elemental function `my_function` replace the inner loop of the matrix multiplication with a call to this new function.

1. Edit the file `chapter4.c` and change the matrix multiplication call a new function in the inner most loop. The code that has to be added is enclosed in the box

```
void MatrixMul( double a[ N][ N], double b[ N][ N], double c[ N][ N])
{
  int i, j;
  for (i = 0; i < N; i++)
  {
    for (j = 0; j < N; j++)
    {
      #ifdef USE_ARRAY_NOTATION
        c[ i][ j] = __sec_reduce_add( a[ i][:] * b[:][j]);
      #else
        for (k=0; k<N; k++) {
          #ifdef USE_ELEMENTAL
              my_elemental(c[i][j], a[i][k], b[k][j]);
        #else
            c[i][j] += a[i][k] * b[k][j];
        #endif
        #endif
      #endif
    }
  }
}
```

2. Edit the file `chapter4.c` and declare the new elemental function just above the MatrixMul function.

```
__declspec(vector (uniform(a,b)))
double my_elemental (double a, double b)
{
    return a + b;
}
```

2. Build the application:

- Linux

```
make  clean
make  CFLAG = "-O2 -DUSE_ELEMENTAL" TARGET=intel.elem
```

- Windows

```
nmake  clean
nmake  CFLAGS= " /O2 –DUSE_ELEMENTAL" TARGET=intel.elem
```

Run the program **intel.elem.exe**.

Questions

a. Is the code vectorised?

b. How can you prove it?