The Map Pattern

A Short Introduction

Joseph Kehoe[1]

[1] Department of Computing and Networking
Institute of Technology Carlow

CDD101, 2017

Table of Contents

1. Definition

2. Examples

3. Optimisation

4. Related Patterns

Table of Contents

1. Definition
2. Examples
3. Optimisation
4. Related Patterns

Definition

- The Map pattern replicates a function over every element of an index set.
- The function is applied to every element in the set concurrently.
- The index set may be abstract or associated with the elements of a collection.
- The function being replicated is called an *elemental function*.

Related Patterns

Map is used for problems that are *Embarrassingly Parallel*.
Often Combined with other patterns

- Collectives often combined with map
  - Gather
  - Reduction
  - Scan
- Generalisations of Map:
  - Stencil
  - Convolution
  - Recurrance
  - Workpile

Type of Concurrency

- If the elemental function contins no control flow then it is *SIMD*
- If there is control flow it is *SPMD*
- Can also be *SIMT* (*SPMD* on tiled *SIMD* hardware)
- If there are no side effects as a result of the elemental function then it is deterministic
- (This is good because?)

Table of Contents

1. Definition
2. Examples
3. Optimisation
4. Related Patterns

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Example SAXPY

SAXPY - Scaled Vector Addition
For each element i in a vector Y: $Y[i] := A \times X[i] + Y[i]$ where Y and X are Vectors and A is a constant

- Depending on type of vector
- float (single precision) **SAXPY**
- Double **DAXPY**
- Complex float **CAXPY**
- Complex Double **ZAXPY**

The operation has a low arithmetic intensity (measure it!)
This implies it does not scale well (why?)

Serial Implementation

Basic Code

```
void saxpy(int n, float a,float y[], float x[])
{
#pragma omp parallel for
        for (int i=0; i < n; ++i)
        {
                y[i]=a * x[i] + y[i];
        }
}
```

- Tiling will improve scalability

Example Mandelbrot Set

- Set of all points on plane c that do not go to infinity when $z = z^2 + c$ is iterated (for ever)
- z starts out at 0
- It has been shown that if the length of z is greater than 2 then it is guaranteed to diverge

Mandelbrot Implementation

Elemental function

```
int calc(Complex c, int depth)
{
        int count=0;
        Complex z=0;
        for(int i=0;i<depth;++i)
        {
                if (abs(z)>2.0)
                {
                        break;
                }
                z=z*z+c;
                count++;
        }
        return count;
}
```

Mandelbrot Implementation

Main Loop

```
        mandel( int p[][], int row, int col,
                int depth){
#pragma omp parallel for collapse(2)
          for(int i=0;i<row,++i)
          {
        for(int k=0;k<col;++k)
                {
               p[i][k]=calc(Complex(i,k),depth);
                }
          }
        }
```

Table of Contents

1. Definition
2. Examples
3. Optimisation
4. Related Patterns

Sequence of Maps versus Map of Sequences

- A sequence of maps does not scale well (why?)
- To increase arithmetic complexity we must do more work between memory reads so change into a map of sequence (code fusion)
- We load all data at the start of the map
- Keep intermediate results in registers
- Write out final result at end

Table of Contents

1. Definition
2. Examples
3. Optimisation
4. Related Patterns

Related Patterns

Stencil each element reads in from surrounding elements but does not write to them

Convolution Stencil with weigths added to neighbouring elements

Workpile Work can grow dynamically as we are doing map

Divide and Conquor Divide problem into smaller problems until base case can be solved serially