

Módulo 7

Challenge 1 - Localidade e SuperEscalaridade

Problema

Considere que tem um conjunto de dados em vírgula flutuante precisão simples organizado como um cubo tri-dimensional:

```
float Cube[N][N][N];          // cubo tri-dimensional
```

Para cada fatia bidimensional do cubo ($Cube[x][y=0..N-1][z=0..N-1]$) pretende-se calcular a expressão abaixo, sendo os resultados guardados no vector

```
float Res[N];                 // vector Resultados
```

$$Res[x] = \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \frac{Cube[x][y][z] + 10.0}{\sqrt[2]{Cube[x][y][0]}}$$

O código abaixo implementa esta operação.

```
void P07Challenge1 (float *Cube, float *Res, int n) {
    int x, y, z;

    for (x = 0; x < n; ++x) {
        Res[x] = 0.f;
    }
    for (y = 0; y < n; ++y) {
        for (z = 0; z < n; ++z) {
            for (x = 0; x < n; ++x) {

                /*
                 Res[x] += (Cube[x][y][z] + 10.) / sqrtf(Cube[x][y][0]);
                 */
                Res[x] += (Cube[x*N*N + y*N + z] + 10.f) / sqrtf(Cube[x*N*N + y*N]);
            }
        }
    }
}
```

Exercício 1 – Copie o ficheiro `/share/acomp/P07-Challenge.zip`, construa o executável e preencha a primeira linha da tabela abaixo para $N=512$.

Para tal deve escrever

```
qsub -F "512 1" P07Challenge.sh
```

| | | N = 512 | | | |
|------------------|---------------------------------------|-----------|-----------|-----|---------------|
| Versão | Observações | T (ms) | #I (G) | CPI | L1_DCM (M) |
| P07-Challenge1() | Versão inicial não otimizada. | | | | |
| P07-Challenge2() | Rearranjo dos loops (x,y,z) | | | | |
| P07-Challenge3() | Variável local | | | | |
| P07-Challenge4() | Expression simplification | | | | |
| P07-Challenge5() | Loop unrolling | | | | |
| P07-Challenge0() | Versão otimizada pela equipa docente. | | | | |

Exercício 2 – Pretende-se que aplicando as técnicas estudadas ao longo do semestre optimize o código acima reduzindo o tempo de execução.

As técnicas a aplicar incluem:

- otimização da localidade nos acessos à memória;
- redução do número de instruções a executar através da simplificação de expressões;
- redução do número de instruções a executar através *do loop unrolling*;
- exploração do *ILP* através da superescalaridade.

Fica portanto vedada a utilização de otimizações relacionadas com processamento vectorial ou utilização de mais do que um núcleo de processamento (*multicore*).

A versão otimizada desenvolvida pela equipa docente pode ser executada com o comando:

```
qsub -F "512 0" P07Challenge.sh
```

Use a tabela acima para anotar os resultados que for obtendo com cada versão da função. Pretende-se essencialmente que aplique o seu espírito crítico na análise dos resultados obtidos.

Para desenhar P07-Challenge4() note que:

$$Res[x] = \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} \frac{Cube[x][y][z] + 10.0}{\sqrt[N]{Cube[x][y][0]}} = \sum_{y=0}^{N-1} \left[\frac{1.0}{\sqrt[N]{Cube[x][y][0]}} * \left(10.0 * N + \sum_{z=0}^{N-1} Cube[x][y][z] \right) \right]$$