

Módulo 12

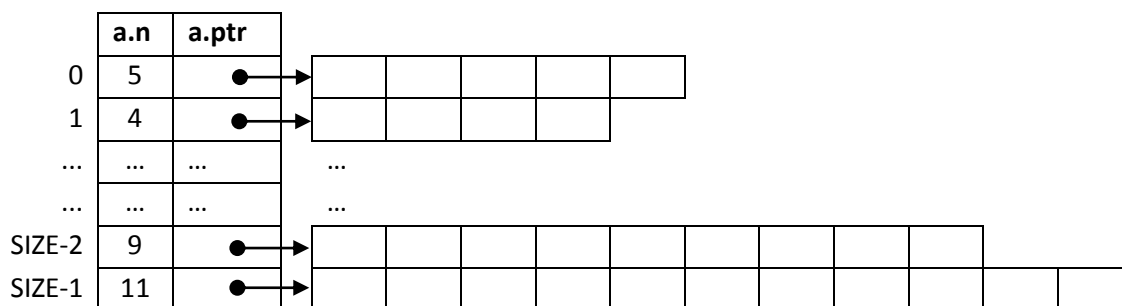
OpenMP e Vectorização - Desafio

Copie o ficheiro `/share/acomp/P12.zip` para a sua pasta, faça o `unzip` e edite `p12.c`.

Note que a função principal invoca 3 funções:

- `initialize()` que cria e inicializa a estrutura de dados descrita abaixo;
- `process()` que processa os dados e **sobre a qual se pretende que aplique as otimizações de vectorização e *multithreading***;
- `release()` que liberta a memória alocada na fase de inicialização.

A estrutura de dados utilizada é um vector de tamanho fixo (dado por `SIZE`). Cada elemento deste vector tem um campo (`ptr`) que aponta para um outro vector com elementos do tipo de dados *float*. Uma vez que o número de elementos de cada um dos vectores de *floats* é variável, cada elemento do vector principal guarda também o número de elementos do vector de *floats* que lhe está associado (campo `n`). A figura abaixo ilustra essa estrutura de dados.



Note que embora o tamanho dos vectores de *floats* varie ao longo do vector principal, e que esse tamanho seja mesmo determinado aleatoriamente, há uma tendência para que o tamanho (dos vectores de *floats*) cresça com o índice do vector principal. Isto é, os elementos de maior índice do vector principal apontarão, com maior probabilidade, para vectores com mais *floats*.

A função `process()` processa cada um destes *floats*. Pretende-se que aplique optimizações relacionadas com vectorização e *multithreading* para reduzir o tempo de execução desta função, preenchendo a tabela abaixo. Compile usando o comando:

```
gcc -O3 -fopenmp -fopenmp-libs -fopenmp-vec -march=ivybridge PL12.c -o PL12
```

Versão	Tempo (ms)	Ganho
Original		----
OpenMP (<i>static scheduling</i>)		
OpenMP (<i>dynamic scheduling</i>)		
Vectorização		
Vectorização + OpenMP		

SUGESTÃO: A operação realizada sobre cada elemento dos vectores de *float* inclui uma dependência RAW:

```
float aux = a[i].ptr[j];
a[i].ptr[j] = aux * aux * aux + 10.0 / aux + 120.0 / (aux * aux) + a[i].ptr[j-1] ;
```

Separar a operação dependente das operações independentes em dois ciclos diferentes consecutivos poderá permitir vectorizar parte desta expressão.